



**Célia Teresa Ligeiro
Mendes Pereira**

Computação no Cálculo das Variações



**Célia Teresa Ligeiro
Mendes Pereira**

Computação no Cálculo das Variações

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática, realizada sob a orientação científica do Doutor João Pedro Antunes Ferreira da Cruz, Professor Auxiliar e Doutor Delfim Fernando Marado Torres, Professor Associado, ambos do Departamento de Matemática da Universidade de Aveiro.

o júri

presidente

Vitor Manuel Carvalho das Neves
Professor Associado da Universidade de Aveiro

Maria Isabel Rocha Ferreira Caiado
Professora Auxiliar da Universidade do Minho

João Pedro Antunes Ferreira da Cruz
Professor Auxiliar da Universidade da Aveiro (orientador)

Delfim Fernando Marado Torres
Professor Associado da Universidade de Aveiro (co-orientador)

agradecimentos

Agradeço

- aos meus orientadores (João Pedro Antunes Ferreira da Cruz e Delfim Fernando Marado Torres) pelas sugestões, acompanhamento e disponibilidade durante todo o processo de elaboração desta dissertação;
- à minha família e aos meus amigos, não só pelo carinho e paciência, mas também por permitirem as minhas frequentes ausências;
- à ESTGOH pelo apoio.

palavras-chave

Cálculo das Variações, Equações de Euler-Lagrange, Discretização, Ferramentas de Cálculo, Braquistócrona, Fenómeno de Lavrentiev

resumo

Nesta dissertação é proposta uma implementação (no *Mathematica* 6) de um algoritmo, desenvolvido por Gusein Guseinov, que determina aproximações da solução óptima de problemas do Cálculo das Variações através da sua discretização. São apresentadas várias discretizações para estes problemas e respectivas equações de Euler-Lagrange. Mostram-se ainda diferentes formulações para problemas do Cálculo das Variações.

Os resultados obtidos pela aplicação do método proposto a duas discretizações do Problema da Braquistócrona e ao Exemplo de Manià (que exhibe o Fenómeno de Lavrentiev) são, ainda, comparados com as soluções encontradas por outras ferramentas de cálculo e por outros métodos.

keywords

Variational Calculus, Calculus of Variations, Euler-Lagrange Equations, Discretization, Solvers, Brachistochrone, Lavrentiev Phenomenon

abstract

This dissertation proposes an implementation (using *Mathematica* 6) of an algorithm, developed by Gusein Guseinov, which determines approximations to the optimal solution of Calculus of Variations problems using discretization. It presents several discretizations for these problems and its Euler-Lagrange equations. It shows different formulations for Calculus of Variations problems. The results obtained applying the proposed method to two discretizations of the Brachistochrone Problem and Manià Example (which shows the Lavrentiev Phenomenon) are compared with the solutions found by other methods and solvers.

Conteúdo

1	Cálculo das Variações Discreto	1
1.1	Introdução	1
1.2	Equações de Euler-Lagrange	2
1.3	Conclusão	23
2	Formulações e Ferramentas Computacionais	25
2.1	Introdução	25
2.2	Diferentes formulações do problema	26
2.3	Ferramentas Computacionais	31
2.4	Algoritmo evolucionista	33
2.5	Conclusão	34
3	Implementação e Comparação	37
3.1	Introdução	37
3.2	Algoritmo sob investigação	38
3.3	Exemplo 1: Problema da Braquistócrona	42
3.3.1	Apresentação do Problema	42
3.3.2	Resultados	48
3.3.3	Estudo Comparativo	54
3.4	Exemplo 2: Problema com Fenómeno de Lavrentiev	57
3.4.1	Apresentação do Problema	57
3.4.2	Resultados	61
3.4.3	Estudo Comparativo	66
3.5	Conclusão	72
4	Conclusão	75
	Anexos	79
A	Cálculos em <i>Maple</i>	79

A.1	Capítulo 1, Exemplo 2	79
A.2	Capítulo 1, Exemplo 1 usando <i>VariationalCalculus</i>	81
B	Implementação no <i>Mathematica</i> 6.0	83
B.1	Problema da Braquistócrona	83
B.2	Problema com Fenómeno de Lavrentiev	92
C	Código C++ usado na ferramenta OC	103
C.1	Problema da Braquistócrona	103
C.2	Problema com Fenómeno de Lavrentiev	109
D	Soluções encontradas pela ferramenta OC	115
D.1	Problema da Braquistócrona	115
D.2	Problema que exhibe o Fenómeno de Lavrentiev	118
	Bibliografia	121
	Índice Remissivo	123

Lista de Tabelas

3.1	Valores de $\int_{x_a}^{x_b} f(x, \bar{y}(x), \bar{y}'(x))dx$ para o problema 3.1 em que $x_a = 0$, $x_b = 10$, $y_a = 10$ e $y_b = 0$	50
3.2	Valores dos integrais das soluções encontradas pela ferramenta OC para $n = 20$	56
3.3	Valores de $\int_{x_a}^{x_b} L(x, \bar{y}(x), \bar{y}'(x))dx$ para o problema em que $x_a = 0$, $x_b = 1$, $y_a = 0$ e $y_b = 1$	62
3.4	Valores dos integrais das soluções encontradas pela ferramenta OC (página 32) para $n = 20$	67
3.5	Valores dos integrais das curvas construídas a partir das soluções encontradas pelos processos descritos.	72
D.1	Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).	115
D.2	Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).	116
D.3	Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).	117
D.4	Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).	118
D.5	Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).	119

Lista de Figuras

1.1	Gráfico que mostra as três curvas (1.13) (linha), (1.11) (ponteada) e (1.12) (tracejada).	22
3.1	Gráfico da curva de tempo mínimo entre os pontos $(0, 10)$ e $(10, 0)$	49
3.2	Curva encontrada aplicando a discretização de Guseinov, com $n = 3$ e o método <i>Nelder Mead</i>	51
3.3	Curva encontrada aplicando a discretização usual, com $n = 3$ e o método <i>Random Search</i>	51
3.4	Curva encontrada aplicando a discretização de Guseinov, com $n = 5$ e o método <i>Random Search</i>	52
3.5	Curva encontrada aplicando a discretização usual, com $n = 5$ e o método <i>Random Search</i>	52
3.6	Curva encontrada aplicando a discretização de Guseinov, com $n = 8$ e o método <i>Random Search</i>	52
3.7	Curva encontrada aplicando a discretização usual, com $n = 8$ e o método <i>Random Search</i>	52
3.8	Curva encontrada aplicando a discretização de Guseinov, com $n = 10$ e o método <i>Nelder Mead</i>	52
3.9	Curva encontrada aplicando a discretização usual, com $n = 10$ e o método <i>Random Search</i>	52
3.10	Curva encontrada aplicando a discretização usual, com $n = 15$ e o método <i>Random Search</i>	53
3.11	Curva encontrada aplicando a discretização de Guseinov, com $n = 20$ e o método <i>Random Search</i>	53
3.12	Curva encontrada aplicando a discretização usual, com $n = 20$ e o método <i>Random Search</i>	53
3.13	Gráfico da melhor solução encontrada pela ferramenta OC (verde) e gráfico da solução óptima (vermelho).	55
3.14	Gráfico da Solução Óptima do exemplo proposto por Manià.	58

3.15	Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 5$ e o método <i>Nelder Mead</i>	61
3.16	Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 5$ e o método <i>Nelder Mead</i>	61
3.17	Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 8$ e o método <i>Differential Evolution</i>	63
3.18	Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 8$ e o método <i>Simulated Annealing</i>	63
3.19	Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 10$ e o método <i>Differential Evolution</i>	63
3.20	Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 10$ e o método <i>Random Search</i>	63
3.21	Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 15$ e o método <i>Differential Evolution</i>	63
3.22	Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 15$ e o método <i>Random Search</i>	63
3.23	Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 20$ e o método <i>Simulated Annealing</i>	64
3.24	Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 20$ e o método <i>Simulated Annealing</i>	64
3.25	Gráfico com soluções encontradas pelo método de truncatura.	66
3.26	Solução obtida pelos métodos “Pesquisa Unidireccional”, Euler, com <i>Piecewise Linear</i> que coincide com a solução obtida pelos métodos “Pesquisa Unidireccional”, Euler, com <i>Piecewise Constant</i>	68
3.27	Solução obtida pelos métodos “Pesquisa Unidireccional”, Runge-Kutta, com <i>Piecewise Linear</i>	68
3.28	Solução obtida pelos métodos “Pesquisa Unidireccional”, Runge-Kutta, com <i>Piecewise Constant</i>	68
3.29	Solução obtida pelos métodos Pesquisa Directa 1, Euler, com <i>Piecewise Linear</i> que coincide com a solução obtida pelos métodos Pesquisa Directa 1, Euler, com <i>Piecewise Constant</i>	68
3.30	Solução obtida pelos métodos Pesquisa Directa 1, Runge-Kutta, com <i>Piecewise Linear</i>	69
3.31	Solução obtida pelos métodos Pesquisa Directa 1, Runge-Kutta, com <i>Piecewise Constant</i>	69
3.32	Solução obtida pelos métodos Pesquisa Directa 2, Euler, com <i>Piecewise Linear</i> que coincide com a solução obtida pelos métodos Pesquisa Directa 2, Euler, com <i>Piecewise Constant</i>	69

3.33	Solução obtida pelos métodos Pesquisa Directa 2, Runge-Kutta, com <i>Piecewise Linear</i>	69
3.34	Solução obtida pelos métodos Pesquisa Directa 2, Runge-Kutta, com <i>Piecewise Constant</i>	69
3.35	Solução obtida pelos métodos Pesquisa Aleatória, Euler, com <i>Piecewise Linear</i>	70
3.36	Solução obtida pelos métodos Pesquisa Aleatória, Euler, com <i>Piecewise Constant</i>	70
3.37	Solução obtida pelos métodos Pesquisa Aleatória, Runge-Kutta, com <i>Piecewise Linear</i>	70
3.38	Solução obtida pelos métodos Pesquisa Aleatória, Runge-Kutta, com <i>Piecewise Constant</i>	70
3.39	Solução obtida pelos métodos Gradientes Conjugados, Euler, com <i>Piecewise Linear</i> que coincide com a solução obtida pelos métodos Gradientes Conjugados, Euler, com <i>Piecewise Constant</i>	70
3.40	Solução obtida pelos métodos Gradientes Conjugados, Runge-Kutta, com <i>Piecewise Linear</i>	71
3.41	Solução obtida pelos métodos Gradientes Conjugados, Runge-Kutta, com <i>Piecewise Constant</i>	71
4.1	Solução óptima do Problema de Resistência Mínima (vermelho) e curva definida pela solução encontrada usando a Discretização de Guseinov, $n = 3$ e método <i>Random Search</i> (azul).	77
4.2	Solução óptima do Problema de Resistência Mínima (vermelho) e curva definida pela solução encontrada usando a Discretização Usual, $n = 20$ e método <i>Nelder Mead</i> (azul).	77

Capítulo 1

Cálculo das Variações Discreto

1.1 Introdução

O Cálculo de Variações resolve (ou tenta resolver) problemas de otimização cujas incógnitas são funções e não variáveis reais. Proporciona “técnicas” que permitem determinar funções que minimizam ou maximizam um funcional.

Esta área da Matemática surgiu (pouco depois do Cálculo Diferencial) a partir de um problema proposto por Johann Bernoulli em 1696 no *Acta Eruditorum Lipsiae* - o Problema da Braquistócrona que consiste em determinar a curva de tempo mínimo que liga dois pontos fixos do mesmo plano vertical.

Este e outros problemas de mecânica (como, por exemplo, determinar formas de veículos que minimizam a resistência ao ar) e de otimização já tinham sido estudados por diversos homens da ciência (Galileu, por exemplo) e ajudaram, através das novas abordagens e usando o Cálculo Diferencial, a desenvolver o Cálculo das Variações. Por outro lado, a resolução de alguns destes problemas através do Cálculo das Variações ajudou a formular e desenvolver algumas leis da Física na área da mecânica, da dinâmica de fluídos, entre outras.

Além disto, o estudo do Cálculo das Variações serviu ainda para desenvolver a Programação Não Linear.

Actualmente o Cálculo das Variações é usado em diferentes áreas da Matemática e da Física, bem como em Biologia, Economia, entre outros.

Apresenta-se, de seguida, a forma geral de um Problema do Cálculo das Variações e algumas abordagens para a sua resolução, terminando-se com o método que será implementado em computador e que resulta da discretização do problema.

1.2 Equações de Euler-Lagrange

A forma geral do **Problema de Cálculo de Variações** apresenta-se de seguida.

Sejam x_a, x_b, α, β números reais em que $x_a < x_b$ e $f : [x_a, x_b] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ (Lagrangiano) é uma função contínua e diferenciável com continuidade.

Considera-se o problema

$$\mathcal{L}[y(\cdot)] = \int_{x_a}^{x_b} f(x, y(x), y'(x)) dx \rightarrow \min \quad (1.1)$$

tal que $y(x_a) = \alpha$ e $y(x_b) = \beta$ (condições de fronteira).

Pretende-se assim determinar funções y , continuamente diferenciáveis em $[x_a, x_b]$, que verificam as condições de fronteira e minimizam o funcional dado em (1.1).

Abordagem Euler

Leonhard Euler (1707 - 1783), no seu *Methodus inveniendi lineas curvas, maximi minimive proprietate gaudentes* (“Método para encontrar linhas curvas que verificam as condições de máximo ou mínimo”, 1744), resolveu problemas do Cálculo das Variações da forma

$$\min_y \int_{x_a}^{x_b} f(x, y(x), y'(x)) dx,$$

discretizando as funções y em vários pontos (x) do domínio (em número finito).

Tem-se, assim, um problema de dimensão finita que é possível resolver usando conhecimentos elementares de análise.

O método usado por Euler assemelha-se, em certa forma, ao método computacional ([4]) que será introduzido em §3.2 e consta no seguinte:

1. Subdividir o intervalo $[x_a, x_b]$ em n subintervalos de igual amplitude Δx :

$$\Delta x = \frac{x_b - x_a}{n},$$

$$[x_a, x_b] = [x_a, x_1] \cup [x_1, x_2] \cup \dots \cup [x_{n-2}, x_{n-1}] \cup [x_{n-1}, x_b] = \bigcup_{i=1}^n [x_{i-1}, x_i], \text{ sendo } x_0 = x_a, x_n =$$

x_b e $x_i = x_0 + i\Delta x$ com $i = 1, 2, \dots, n-1$.

2. Aproximar y' , em cada intervalo, pelo declive da recta que liga os pontos extremos do intervalo: $y'(x_i) = \frac{y(x_i) - y(x_{i-1})}{\Delta x}$.

3. Aproximar o funcional (denota-se $y(x_i) = y_i$ para tornar mais clara a expressão):

$$\begin{aligned} \int_{x_a}^{x_b} f(x, y(x), y'(x)) dx &\approx \sum_{i=1}^n f\left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x}\right) \Delta x \\ &= f\left(x_1, y_1, \frac{y_1 - y_0}{\Delta x}\right) \Delta x + \dots + f\left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x}\right) \Delta x \\ &\quad + f\left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x}\right) \Delta x + \dots + f\left(x_n, y_n, \frac{y_n - y_{n-1}}{\Delta x}\right) \Delta x \\ &= I(y_0, y_1, \dots, y_n), \end{aligned}$$

sendo I uma função de $n + 1$ variáveis independentes).

4. Aplicar a Condição Necessária de Optimalidade $\frac{\partial I}{\partial y}(y_0, y_1, \dots, y_n) = 0$:

Para cada $y_i, i = 1, \dots, n - 1$

$$\begin{aligned} \frac{\partial I}{\partial y_i}(y_0, y_1, \dots, y_n) &= \frac{\partial}{\partial y_i} \left(f(x_1, y_1, \frac{y_1 - y_0}{\Delta x}) \Delta x \right) + \dots + \frac{\partial}{\partial y_i} \left(f(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x}) \Delta x \right) \\ &\quad + \frac{\partial}{\partial y_i} \left(f(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x}) \Delta x \right) + \dots + \frac{\partial}{\partial y_i} \left(f(x_n, y_n, \frac{y_n - y_{n-1}}{\Delta x}) \Delta x \right) \\ &= \frac{\partial}{\partial y_i} \left(f(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x}) \Delta x \right) + \frac{\partial}{\partial y_i} \left(f(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x}) \Delta x \right) \\ &= \frac{\partial f}{\partial y_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) \Delta x + \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) - \frac{\partial f}{\partial y'_i} \left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right). \end{aligned}$$

Tem-se então, para $i = 1, \dots, n - 1$,

$$\begin{aligned} &\frac{\partial f}{\partial y_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) \Delta x + \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) - \frac{\partial f}{\partial y'_i} \left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) = 0 \\ \Leftrightarrow &\quad \frac{\partial f}{\partial y_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) \Delta x = \frac{\partial f}{\partial y'_i} \left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) - \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) \\ \Leftrightarrow &\quad \frac{\partial f}{\partial y_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) = \frac{\frac{\partial f}{\partial y'_i} \left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) - \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right)}{\Delta x} \\ \Leftrightarrow &\quad \frac{\partial f}{\partial y_i} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) = \frac{\Delta \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x} \quad (\text{ver equação no Teorema 1.2.2, página 8}) \quad (1.2) \end{aligned}$$

5. Quando o número de intervalos da partição de $[x_a, x_b]$ tende para o infinito, tem-se $\Delta x \longrightarrow 0$:

$$\lim_{\Delta x \longrightarrow 0} \frac{\frac{\partial f}{\partial y'_i} \left(x_{i+1}, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) - \frac{\partial f}{\partial y'_i} \left(x_i, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right)}{\Delta x} = \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y, y') \right).$$

Assim, obtêm-se as famosas *equações de Euler* (mais tarde conhecidas por *equações de Euler-Lagrange*):

$$\frac{\partial f}{\partial y}(x, y(x), y'(x)) = \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right). \quad (1.3)$$

Tendo em conta a formulação do problema, a *Equação de Euler-Lagrange* vai resultar numa equação diferencial.

Abordagem de Lagrange

A técnica proposta por Lagrange para resolver o *Problema da Braquistócrona* “perturba” ou “faz variar” y . A mesma técnica de “variação” permite uma demonstração alternativa das equações (1.3) de Euler.

Já se viu que se pretende encontrar y que minimiza $\int_{x_a}^{x_b} f(x, y(x), y'(x)) dx = I[y]$.

Começa-se por considerar $Y(x) = y(x) + \varepsilon\eta(x)$, em que η é uma função diferenciável qualquer com $\eta(x_a) = \eta(x_b) = 0$ (η designa-se por *variação*) e ε um valor real.

Desta forma, Y é admissível:

$$\begin{aligned} Y(x_a) &= y(x_a) + \varepsilon\eta(x_a) = y(x_a) = y_a = \alpha \text{ e} \\ Y(x_b) &= y(x_b) + \varepsilon\eta(x_b) = y(x_b) = y_b = \beta. \end{aligned}$$

Ou seja, são verificadas as condições requeridas (Y é uma função diferenciável que liga os pontos (x_a, α) e (x_b, β)).

Para qualquer função $\eta(x)$ é possível definir um intervalo suficientemente pequeno para ε de forma que $Y(x)$ e $y(x)$ estejam tão próximos quanto se queira.

Pode, assim, definir-se o integral em função de ε :

$$I(\varepsilon) = \int_{x_a}^{x_b} f(x, Y(x), Y'(x)) dx,$$

$$Y'(x) = y'(x) + \varepsilon\eta'(x).$$

Se y minimiza I , então $I[y] \leq I[y + \varepsilon\eta]$.

Fazendo $\varepsilon = 0$ obtêm-se $Y(x) = y(x)$ e $I(0) = I[y]$. Como y minimiza I então I , enquanto função da variável ε , tem um mínimo em $\varepsilon = 0$.

Usando a condição necessária, $I'(0) = 0$, para encontrar possíveis extremantes, obtém-se:

$$\begin{aligned} I'(\varepsilon) &= \frac{dI}{d\varepsilon}(\varepsilon) = \int_{x_a}^{x_b} \frac{d}{d\varepsilon} f(x, Y(x), Y'(x)) dx \\ &= \int_{x_a}^{x_b} \left(\frac{\partial f}{\partial Y}(x, Y(x), Y'(x)) \frac{\partial Y}{\partial \varepsilon}(x) + \frac{\partial f}{\partial Y'}(x, Y(x), Y'(x)) \frac{\partial Y'}{\partial \varepsilon}(x) \right) dx \\ &= \int_{x_a}^{x_b} \left(\frac{\partial f}{\partial Y}(x, Y(x), Y'(x)) \eta(x) + \frac{\partial f}{\partial Y'}(x, Y(x), Y'(x)) \eta'(x) \right) dx. \end{aligned}$$

Para $\varepsilon = 0$, tem-se

$$\begin{aligned} I'(0) &= \int_{x_a}^{x_b} \left(\frac{\partial f}{\partial y}(x, y(x), y'(x)) \eta(x) + \frac{\partial f}{\partial y'}(x, y(x), y'(x)) \eta'(x) \right) dx \\ &= \int_{x_a}^{x_b} \frac{\partial f}{\partial y}(x, y(x), y'(x)) \eta(x) dx + \int_{x_a}^{x_b} \frac{\partial f}{\partial y'}(x, y(x), y'(x)) \eta'(x) dx. \end{aligned}$$

Usando *Integração por Partes*, pode escrever-se

$$\begin{aligned}
I'(0) &= \int_{x_a}^{x_b} \frac{\partial f}{\partial y}(x, y(x), y'(x)) \eta(x) dx + \left[\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \eta(x) \right]_{x_a}^{x_b} \\
&\quad - \int_{x_a}^{x_b} \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right) \eta(x) dx \\
&= \int_{x_a}^{x_b} \frac{\partial f}{\partial y}(x, y(x), y'(x)) \eta(x) dx - \int_{x_a}^{x_b} \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right) \eta(x) dx \\
&= \int_{x_a}^{x_b} \left(\frac{\partial f}{\partial y}(x, y(x), y'(x)) - \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right) \right) \eta(x) dx.
\end{aligned}$$

Tem-se assim que

$$\int_{x_a}^{x_b} \left(\frac{\partial f}{\partial y}(x, y(x), y'(x)) - \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right) \right) \eta(x) dx = 0.$$

Usando o lema seguinte (demonstração, e.g., em [13]),

Lema 1.2.1. *Dados x_1 e x_2 dois valores reais fixos e uma função G contínua em $[x_1, x_2]$, se*

$$\int_{x_2}^{x_1} G(x) \eta(x) dx = 0$$

para qualquer função continuamente diferenciável η que verifica $\eta(x_1) = \eta(x_2) = 0$, então $G(x) = 0$, $\forall x \in [x_1, x_2]$.

obtém-se, de novo, a equação (1.3):

$$\frac{\partial f}{\partial y}(x, y(x), y'(x)) - \frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y(x), y'(x)) \right) = 0$$

(*equação de Euler-Lagrange*).

Qualquer função y que verifique a *equação de Euler-Lagrange* é, por definição, extremal (“função crítica”) de

$$\int_{x_a}^{x_b} f(x, y(x), y'(x)) dx,$$

ou seja, é uma função que minimiza, maximiza ou tem um papel semelhante a um ponto de sela.

Abordagem Discreta

O problema do Cálculo das Variações (1.1) foi apresentado, na Secção 1.2, na forma contínua. No entanto, o tratamento computacional e numérico de qualquer problema é discreto, isto é, não são considerados conjuntos não enumeráveis (intervalos) mas sequências

finitas de pontos que aproximam os intervalos. Espera-se que quanto maior for o número de pontos mais próximo estará a solução discretizada da solução contínua.

Em [10] (páginas 162–165) é demonstrado que a solução discreta de um problema de Controlo Óptimo converge para a solução contínua. Como os problemas do Cálculo das Variações são casos particulares dos problemas de Controlo Óptimo (§2.2, página 26), pode concluir-se também que as aproximações discretas do Cálculo das Variações convergem para a solução contínua do problema.

No entanto, para o mesmo problema podem ser escolhidos, por exemplo, diferentes pontos ou pontos em diferente número. Apresentam-se a seguir diferentes formas de discretizar um problema do Cálculo das Variações.

“Discretização Usual” – Diferenças Finitas de Euler

Já foi referido o método usado por Euler (§1.2, página 2) para resolver o Problema da Braquistócrona (§3.3.1, página 42). Este método considera o problema na forma discreta procurando os valores dos y_i ($i = 1, \dots, n-1$) que minimizam a expressão

$$\sum_{i=1}^n f\left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x}\right) \Delta x, \quad (1.4)$$

sabendo que $y_0 = y_a = \alpha$ e $y_n = y_b = \beta$.

Foram também encontradas as equações de Euler-Lagrange ((1.2) na página 3) para este problema (1.4):

$$\frac{\partial f}{\partial y}\left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x}\right) = \frac{\Delta \frac{\partial f}{\partial y'}\left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x}\right)}{\Delta x}.$$

“Discretização à la Smirnov-Bushenkov”

Smirnov e Bushenkov ([10]) discretizam os problemas do Cálculo das Variações usando o método anterior (intervalos de igual amplitude) mas adaptam o problema (por intermédio de uma mudança de variável apropriada) de forma a terem como domínio de integração o intervalo $[0, 1]$.

Para isso é efectuada a mudança de variável $t = \frac{x-x_a}{x_b-x_a}$, ou seja, $x = (x_b - x_a)t + x_a$. Assim, como $x \in [x_a, x_b]$, tem-se que $t \in [0, 1]$. Além disso, $dt = \frac{1}{x_b-x_a}dx$.

Logo,

$$\begin{aligned} \int_{x_a}^{x_b} f(x, y(x), y'(x))dx &= \\ &= \int_0^1 f((x_b - x_a)t + x_a, y((x_b - x_a)t + x_a), y'((x_b - x_a)t + x_a)) (x_b - x_a)dt. \end{aligned}$$

Pretende-se então encontrar a função y que minimiza o funcional

$$\int_0^1 f((x_b - x_a)t + x_a, y((x_b - x_a)t + x_a), y'((x_b - x_a)t + x_a)) dt$$

sob as condições de fronteira $y(0) = \alpha$ e $y(1) = \beta$.

A partir daqui pode aplicar-se a discretização anterior (Diferenças Finitas de Euler).

“Discretização de Guseinov”

Na abordagem discreta para a resolução deste problema, discretiza-se o domínio de integração, considerando-se um conjunto finito de números reais

$$X = \{x_0, x_1, \dots, x_n\},$$

em que $x_a = x_0 < x_1 < \dots < x_n = x_b$.

A partir de cada função y (cujos pontos são determinados na resolução do problema)

$$\begin{aligned} y : X &\longrightarrow \mathbb{R} \\ x_0 &\mapsto y(x_0) = y_0 = \alpha \\ x_i &\mapsto y(x_i) = y_i \\ x_n &\mapsto y(x_n) = y_n = \beta \end{aligned}$$

é possível definir uma linha poligonal que liga os pontos (x_0, α) e (x_n, β) , passando pelos pontos (x_i, y_i) , com $i = 1, \dots, n-1$.

Pode assim definir-se um conjunto F de funções admissíveis (funções que verificam as condições anteriores):

$$F = \{y : X \longrightarrow \mathbb{R} \text{ tal que } y(x_0) = \alpha \text{ e } y(x_n) = \beta\}.$$

Interessa ainda conhecer o comportamento do funcional contínuo f em cada intervalo $[x_{i-1}, x_i]$.

Considere-se uma função

$$\begin{aligned} L : X \times X \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} &\longrightarrow \mathbb{R} \\ (s, t, u, v, w) &\mapsto L(s, t, u, v, w) \end{aligned}$$

cujas derivadas parciais de primeira ordem (em relação a cada um dos últimos três argumentos) são contínuas ($\frac{\partial L}{\partial u}, \frac{\partial L}{\partial v}, \frac{\partial L}{\partial w} \in C^0[\mathbb{R}]$).

As variáveis s e t dizem respeito ao domínio do funcional (1.1), $u = y(s)$, $v = y(t)$ e $w = \frac{v-u}{t-s}$.

Assim, o funcional que se pretende minimizar (para maximizar o processo é análogo) tem a forma (discreta)

$$J[y] = \sum_{i=1}^n L\left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right) \Delta x_{i-1}. \quad (1.5)$$

Ou seja, procuram-se funções y de F que minimizem (ou maximizem) J .

Há duas diferenças fundamentais entre esta discretização e as restantes.

A primeira diferença diz respeito aos pontos x escolhidos. No método das Diferenças Finitas Δx é constante, isto é, $\Delta x = x_1 - x_0 = x_2 - x_1 = \cdots = x_i - x_{i-1} = \cdots = x_n - x_{n-1} = \frac{x_b - x_a}{n}$ enquanto que nesta discretização considera-se $\Delta x_{i-1} = x_i - x_{i-1}$, não necessariamente constante.

A segunda diferença tem a ver com o funcional. O método de discretização usual considera, como já se viu, o mesmo funcional na versão discreta.

$$\min_{y(x)} \int_{x_a}^{x_b} f(x, y(x), y'(x)) \longrightarrow \min_{y(x)} \sum_{i=1}^n f\left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x}\right) \Delta x$$

O funcional discreto considerado por Guseinov ([4]) é determinado para cada caso, tendo em conta que uma curva que minimiza ou maximiza o funcional é aproximada por segmentos de recta, uma área é aproximada por áreas de trapézios, ... Assim, só é possível encontrar o funcional discreto de exemplos em que se conheça o seu significado, ajustando a formulação ao exemplo (Exemplo 1 na página 12, Exemplo 2 na página 16 e Problema da Braquistócrona, em §3.3.1, página 43):

$$\min_{y(x)} \int_{x_a}^{x_b} f(x, y(x), y'(x)) \longrightarrow \min_{y(x)} \sum_{i=1}^n L\left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right) \Delta x_{i-1}$$

O próximo teorema introduz a forma das equações de Euler-Lagrange para a formulação de Guseinov.

Teorema 1.2.2 (Equação de Euler-Lagrange Discreta [4]). *Se o problema do Cálculo das Variações tem um extremo local em \hat{y} então \hat{y} verifica a equação de Euler-Lagrange*

$$\begin{aligned} \frac{\partial L}{\partial u}\left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{\Delta y_i}{\Delta x_i}\right) \frac{\Delta x_i}{\Delta x_{i-1}} + \frac{\partial L}{\partial v}\left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right) \\ = \frac{\Delta \frac{\partial L}{\partial w}\left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)}{\Delta x_{i-1}}, \end{aligned} \quad (1.6)$$

para $i \in \{1, 2, \dots, n-1\}$.

Além disso, como $\hat{y} \in F$, então $\hat{y}(x_0) = \alpha$ e $\hat{y}(x_n) = \beta$.

Demonstração. Pretende mostrar-se que para todo o $i \in \{1, 2, \dots, n-1\}$

$$\begin{aligned} \frac{\partial L}{\partial u}\left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i}\right) \Delta x_i + \frac{\partial L}{\partial v}\left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}}\right) \Delta x_{i-1} \\ = \Delta \frac{\partial L}{\partial w}\left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}}\right). \end{aligned}$$

Por hipótese, a função \hat{y} é extremante local de J , em F . Para fazer variar \hat{y} numa vizinhança, considere-se a função $\eta : X \longrightarrow \mathbb{R}$ que verifica as condições $\eta(x_0) = \eta_0 = 0$ e $\eta(x_n) = \eta_n = 0$. Para averiguar o comportamento de J definido em (1.5) considere-se a função

$$\begin{aligned} \varphi : \mathbb{R} &\longrightarrow \mathbb{R} \\ \varepsilon &\longmapsto \varphi(\varepsilon) = J[\hat{y} + \varepsilon\eta] \\ &= \sum_{i=1}^n L \left(x_{i-1}, x_i, \hat{y}_{i-1} + \varepsilon\eta_{i-1}, \hat{y}_i + \varepsilon\eta_i, \frac{\Delta(\hat{y}_{i-1} + \varepsilon\eta_{i-1})}{\Delta x_{i-1}} \right) \Delta x_{i-1}, \end{aligned}$$

$\varepsilon \in \mathbb{R}$. Tem-se $\hat{y} + \varepsilon\eta \in F, \hat{y} + \varepsilon\eta : X \mapsto \mathbb{R}$, pois

- $[\hat{y} + \varepsilon\eta](x_0) = \hat{y}(x_0) + \varepsilon\eta(x_0) = \alpha + 0 = \alpha;$
 $[\hat{y} + \varepsilon\eta](x_n) = \hat{y}(x_n) + \varepsilon\eta(x_n) = \beta + 0 = \beta;$
- $\forall x \in X \quad [\hat{y} + \varepsilon\eta](x) = \hat{y}(x) + [\varepsilon\eta](x) = \hat{y}(x) + \varepsilon\eta(x)$, pelas propriedades de soma de funções e produto de uma constante por uma função.

Por outro lado, $\varphi(0) = J[\hat{y}]$ e, como $J[\hat{y}]$ é extremo local, então $\varphi'(0) = 0$. Derivando a função φ , tem-se que

$$\begin{aligned} \varphi'(\varepsilon) &= \sum_{i=1}^n \frac{d}{d\varepsilon} L \left(x_{i-1}, x_i, \hat{y}_{i-1} + \varepsilon\eta_{i-1}, \hat{y}_i + \varepsilon\eta_i, \frac{\Delta(\hat{y}_{i-1} + \varepsilon\eta_{i-1})}{\Delta x_{i-1}} \right) \Delta x_{i-1} \\ &\quad + \sum_{i=1}^n L \left(x_{i-1}, x_i, \hat{y}_{i-1} + \varepsilon\eta_{i-1}, \hat{y}_i + \varepsilon\eta_i, \frac{\Delta(\hat{y}_{i-1} + \varepsilon\eta_{i-1})}{\Delta x_{i-1}} \right) \frac{d}{d\varepsilon} \Delta x_{i-1} \\ &= \sum_{i=1}^n \frac{d}{d\varepsilon} L \left(x_{i-1}, x_i, \hat{y}_{i-1} + \varepsilon\eta_{i-1}, \hat{y}_i + \varepsilon\eta_i, \frac{\Delta\hat{y}_{i-1} + \varepsilon\Delta\eta_{i-1}}{\Delta x_{i-1}} \right) \Delta x_{i-1} \\ &= \sum_{i=1}^n \left[\frac{\partial L}{\partial u}(\cdots) \frac{du}{d\varepsilon}(\cdots) + \frac{\partial L}{\partial v}(\cdots) \frac{dv}{d\varepsilon}(\cdots) + \frac{\partial L}{\partial w}(\cdots) \frac{dw}{d\varepsilon}(\cdots) \right] \Delta x_{i-1} \\ &= \sum_{i=1}^n \frac{\partial L}{\partial u}(\cdots) \frac{du}{d\varepsilon}(\cdots) \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial v}(\cdots) \frac{dv}{d\varepsilon}(\cdots) \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial w}(\cdots) \frac{dw}{d\varepsilon}(\cdots) \Delta x_{i-1} \\ &= \sum_{i=1}^n \frac{\partial L}{\partial u}(\cdots) \eta_{i-1} \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial v}(\cdots) \eta_i \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial w}(\cdots) \frac{\Delta\eta_{i-1} \Delta x_{i-1}}{(\Delta x_{i-1})^2} \Delta x_{i-1} \\ &= \sum_{i=1}^n \frac{\partial L}{\partial u}(\cdots) \eta_{i-1} \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial v}(\cdots) \eta_i \Delta x_{i-1} + \sum_{i=1}^n \frac{\partial L}{\partial w}(\cdots) \Delta\eta_{i-1}, \end{aligned}$$

onde $(\cdots) = \left(x_{i-1}, x_i, \hat{y}_{i-1} + \varepsilon\eta_{i-1}, \hat{y}_i + \varepsilon\eta_i, \frac{\Delta\hat{y}_{i-1} + \varepsilon\Delta\eta_{i-1}}{\Delta x_{i-1}} \right)$.

A condição necessária de optimalidade $\varphi'(0) = 0$ toma a seguinte forma:

$$\begin{aligned}
& \sum_{i=1}^n \frac{\partial L}{\partial u} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \eta_{i-1} \Delta x_{i-1} \\
& + \sum_{i=1}^n \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \eta_i \Delta x_{i-1} \\
& + \sum_{i=1}^n \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \Delta \eta_{i-1} = 0 \\
\Leftrightarrow & \sum_{i=0}^{n-1} \frac{\partial L}{\partial u} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_i \Delta x_i \\
& + \sum_{i=0}^{n-1} \frac{\partial L}{\partial v} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_{i+1} \Delta x_i \\
& + \sum_{i=0}^{n-1} \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \Delta \eta_i = 0 \\
\Leftrightarrow & \frac{\partial L}{\partial u} \left(x_0, x_1, \hat{y}_0, \hat{y}_1, \frac{\Delta \hat{y}_0}{\Delta x_0} \right) \eta_0 \Delta x_0 + \sum_{i=1}^{n-1} \frac{\partial L}{\partial u} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_i \Delta x_i \\
& + \sum_{i=0}^{n-2} \frac{\partial L}{\partial v} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_{i+1} \Delta x_i + \frac{\partial L}{\partial v} \left(x_{n-1}, x_n, \hat{y}_{n-1}, \hat{y}_n, \frac{\Delta \hat{y}_n}{\Delta x_n} \right) \eta_n \Delta x_{n-1} \\
& + \sum_{i=0}^{n-1} \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \Delta \eta_i = 0.
\end{aligned}$$

Tendo em conta que $\eta_0 = \eta_n = 0$ e aplicando a *soma por partes*,

$$\begin{aligned}
\sum_{i=0}^{n-1} \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \Delta \eta_i &= - \sum_{i=0}^{n-2} \Delta \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_{i+1} \\
&= - \sum_{i=1}^{n-1} \Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \eta_i,
\end{aligned}$$

obtém-se a equação

$$\begin{aligned}
& \sum_{i=1}^{n-1} \frac{\partial L}{\partial u} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_i \Delta x_i \\
& + \sum_{i=0}^{n-2} \frac{\partial L}{\partial v} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \eta_{i+1} \Delta x_i \\
& - \sum_{i=1}^{n-1} \Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \eta_i = 0
\end{aligned}$$

que é equivalente a

$$\begin{aligned} \Leftrightarrow \sum_{i=1}^{n-1} & \left[\frac{\partial L}{\partial u} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \Delta x_i \right. \\ & + \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \Delta x_{i-1} \\ & \left. - \Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \right] \eta_i = 0. \end{aligned}$$

Como η_i é qualquer número real para $i \in \{1, 2, \dots, n-1\}$, então

$$\begin{aligned} \frac{\partial L}{\partial u} \left(x_i, x_{i+1}, \hat{y}_i, \hat{y}_{i+1}, \frac{\Delta \hat{y}_i}{\Delta x_i} \right) \Delta x_i + \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) \Delta x_{i-1} \\ - \Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, \hat{y}_{i-1}, \hat{y}_i, \frac{\Delta \hat{y}_{i-1}}{\Delta x_{i-1}} \right) = 0. \end{aligned}$$

□

Usando o Teorema 1.2.2 (e os pressupostos introdutórios), foi elaborado por Guseinov ([4]) um (novo) método que, através da discretização do intervalo, permite determinar soluções aproximadas para problemas do Cálculo das Variações.

Apesar de ser aplicada a “Discretização de Guseinov” (página 7) nos exemplos usados na apresentação do método ([4]), no Teorema 1.2.2 não há qualquer indicação acerca do *tipo* de funcional discreto L . L deve, apenas, verificar as condições $\frac{\partial L}{\partial u}, \frac{\partial L}{\partial v}, \frac{\partial L}{\partial w} \in C^0[\mathbb{R}]$. Assim, pode aplicar-se o método usando “Discretização Usual” desde que o funcional discreto (1.5) usual verifique tais condições.

As aproximações de soluções encontradas são candidatas a extremantes (*funções críticas*), isto é, podem minimizar, maximizar ou nem maximizar nem minimizar (*funções sela*) o funcional. Para averiguar se uma extremal encontrada é maximizante ou minimizante deve recorrer-se às derivadas de segunda ordem. No entanto, tendo em conta o número de variáveis envolvidas no problema, os cálculos necessários para permitir tirar conclusões a este respeito são elevados. Mesmo o ajuste da Condição (necessária) de Legendre¹ ([12]) a esta formulação discreta, que poderia excluir a possibilidade da aproximação encontrada ser minimizante ou maximizante, acarreta muitos cálculos².

¹Condição de Legendre:

Se \hat{y} é minimizante local do problema do Cálculo das Variações (1.1) então

$$\frac{\partial f^2}{\partial y'^2} (x, \hat{y}(x), \hat{y}'(x)) \geq 0$$

para todo o $x \in [x_a, x_b]$.

²Em [6] são apresentadas condições suficientes e condições necessárias para averiguar que tipo de aproximação se encontrou (minimizante, maximizante ou nenhuma das duas) para a solução de um problema discreto na forma usual com $\Delta x = 1$.

Exemplos

Vejam-se alguns exemplos que aplicam a equação de Euler-Lagrange do Teorema 1.2.2 às diferentes discretizações apresentadas.

Exemplo 1. Para determinar a curva que minimiza a distância entre os pontos $(-1, 2)$ e $(5, 4)$ tem-se o seguinte problema do Cálculo das Variações:

$$\min_y \int_{-1}^5 \sqrt{1 + y'(x)^2} dx =: I[y] \quad (1.7)$$

em que

$$y(-1) = 2 \text{ e } y(5) = 4.$$

Neste caso $f(x, y, y') = \sqrt{1 + y'^2}$. Comece-se por determinar a função L usando a discretização de Guseinov e a discretização usual.

a) Discretização de Guseinov

A curva que minimiza a distância entre dois pontos será aproximada por segmentos de recta que ligam os pontos (x_i, y_i) e (x_{i+1}, y_{i+1}) . Procuram-se os valores y_i de modo que a soma dos comprimentos dos segmentos de recta encontrados seja mínima.

A medida do comprimento do segmento de recta $[(x_i, y_i), (x_{i+1}, y_{i+1})]$ é

$$\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2} = \sqrt{1 + \frac{(\Delta y_i)^2}{(\Delta x_i)^2}} \Delta x_i.$$

Supondo que se pretende discretizar o intervalo $[-1, 5]$ com 4 pontos ($n = 3$) igualmente espaçados, tem-se $X = \{-1, 1, 3, 5\}$.

$$I[y] = \int_{-1}^5 \sqrt{1 + y'(x)^2} dx \approx \sum_{i=0}^2 \sqrt{1 + \left(\frac{y(x_{i+1}) - y(x_i)}{x_{i+1} - x_i} \right)^2} \Delta x_i =: J[y].$$

Define-se a função auxiliar

$$\begin{aligned} L : X \times X \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} &\longrightarrow \mathbb{R} \\ (s, t, u, v, w) &\mapsto L(s, t, u, v, w) = \sqrt{1 + w^2} \end{aligned}$$

Para usar as equações de Euler-Lagrange (1.6) deve ainda determinar-se as derivadas parciais de L :

$$\begin{aligned} L_u &= \frac{\partial L}{\partial u}(s, t, u, v, w) = 0 \\ L_v &= \frac{\partial L}{\partial v}(s, t, u, v, w) = 0 \\ L_w &= \frac{\partial L}{\partial w}(s, t, u, v, w) = \frac{w}{\sqrt{1 + w^2}}. \end{aligned}$$

Consideram-se todos os subintervalos de igual amplitude pelo que $\Delta x_i = \Delta x$ é constante, $\forall i \in \{1, 2\}$. A equação de Euler-Lagrange

$$\frac{\partial L}{\partial u} \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{\Delta y_i}{\Delta x} \right) + \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) = \frac{\Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x}$$

pode ser simplificada tendo em conta que $L_u = L_v = 0$

$$\begin{aligned} & \frac{\Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x} = 0 \\ \Leftrightarrow & \Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) = 0 \\ \Leftrightarrow & \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{\Delta y_i}{\Delta x} \right) - \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) = 0. \end{aligned}$$

Na função L considerada ($\Delta x = 2$)

$$\frac{\frac{y_{i+1}-y_i}{2}}{\sqrt{1 + \left(\frac{y_{i+1}-y_i}{2} \right)^2}} - \frac{\frac{y_i-y_{i-1}}{2}}{\sqrt{1 + \left(\frac{y_i-y_{i-1}}{2} \right)^2}} = 0 \quad (1.8)$$

Concretizando o valor de i para 1 e 2, tem-se que

$i = 1 :$

$$\frac{\frac{y(3)-y(1)}{2}}{\sqrt{1 + \left(\frac{y(3)-y(1)}{2} \right)^2}} - \frac{\frac{y(1)-y(-1)}{2}}{\sqrt{1 + \left(\frac{y(1)-y(-1)}{2} \right)^2}} = 0;$$

$i = 2 :$

$$\frac{\frac{y(5)-y(3)}{2}}{\sqrt{1 + \left(\frac{y(5)-y(3)}{2} \right)^2}} - \frac{\frac{y(3)-y(1)}{2}}{\sqrt{1 + \left(\frac{y(3)-y(1)}{2} \right)^2}} = 0.$$

Estas duas equações e as duas condições de fronteira compõem um sistema de equações não lineares cujas incógnitas são $y(-1)$, $y(1)$, $y(3)$ e $y(5)$:

$$\begin{cases} \frac{\frac{y(3)-y(1)}{2}}{\sqrt{1 + \left(\frac{y(3)-y(1)}{2} \right)^2}} - \frac{\frac{y(1)-y(-1)}{2}}{\sqrt{1 + \left(\frac{y(1)-y(-1)}{2} \right)^2}} = 0 \\ \frac{\frac{y(5)-y(3)}{2}}{\sqrt{1 + \left(\frac{y(5)-y(3)}{2} \right)^2}} - \frac{\frac{y(3)-y(1)}{2}}{\sqrt{1 + \left(\frac{y(3)-y(1)}{2} \right)^2}} = 0 \\ y(-1) = 2 \\ y(5) = 4 \end{cases}.$$

Resolvendo o sistema obtém-se

$$\begin{cases} y(-1) = 2 \\ y(1) = \frac{8}{3} \\ y(3) = \frac{10}{3} \\ y(5) = 4 \end{cases}.$$

Usando diferentes tipos de interpolação é possível encontrar várias funções \bar{y} a partir dos pontos encontrados. Como se está a discretizar um problema contínuo, definir-se-á uma função por ramos lineares cujo gráfico liga os pontos obtidos $((-1, 2), (1, \frac{8}{3}), (3, \frac{10}{3}), (5, 4))$ dois a dois com um segmento de recta:

$$\bar{y}(x) = \begin{cases} \frac{7}{3} + \frac{x}{3} & \text{se } -1 \leq x \leq 1 \\ \frac{7}{3} + \frac{x}{3} & \text{se } 1 \leq x < 3 \\ \frac{7}{3} + \frac{x}{3} & \text{se } 3 \leq x < 5 \end{cases}.$$

Neste caso, obtém-se uma função linear

$$\bar{y}(x) = \frac{7}{3} + \frac{x}{3}, \forall x \in [-1, 5],$$

que é a solução óptima do problema discreto (e também do problema contínuo) proposto (1.7) e tem como valor do funcional

$$\begin{aligned} J[\bar{y}] &= \sum_{i=0}^2 \sqrt{1 + \left(\frac{\bar{y}(x_{i+1}) - \bar{y}(x_i)}{x_{i+1} - x_i} \right)^2} \Delta x_i \\ &= \Delta x \sum_{i=0}^2 \sqrt{1 + \left(\frac{\bar{y}(x_{i+1}) - \bar{y}(x_i)}{\Delta x} \right)^2} \\ &= \Delta x \left(\sqrt{1 + \left(\frac{\bar{y}(x_1) - \bar{y}(x_0)}{\Delta x} \right)^2} + \sqrt{1 + \left(\frac{\bar{y}(x_2) - \bar{y}(x_1)}{\Delta x} \right)^2} + \sqrt{1 + \left(\frac{\bar{y}(x_3) - \bar{y}(x_2)}{\Delta x} \right)^2} \right) \\ &= 2 \left(\sqrt{1 + \left(\frac{\frac{8}{3} - 2}{2} \right)^2} + \sqrt{1 + \left(\frac{\frac{10}{3} - \frac{8}{3}}{2} \right)^2} + \sqrt{1 + \left(\frac{4 - \frac{10}{3}}{2} \right)^2} \right) \\ &= 2\sqrt{10}. \end{aligned}$$

b) Discretização Usual

Usando este tipo de discretização, a determinação de L é imediata:

$$L(s, t, u, v, w) = \sqrt{1 + w^2}.$$

Neste caso, L é igual em ambas as discretizações, pelo que a solução obtida usando a discretização usual coincide com a solução já encontrada pela discretização de Guseinov.

c) Método de Euler

Neste problema o método das Diferenças Finitas de Euler é igual, em cálculos, ao método de Guseinov porque a formulação como problema discreto e as equações de Euler-Lagrange coincidem.

$$\int_{-1}^5 \sqrt{1 + y'(x)^2} dx \approx \sum_{i=0}^2 \sqrt{1 + \left(\frac{\Delta y_i}{\Delta x} \right)^2} \Delta x.$$

Equações de Euler-Lagrange:

$$\frac{\partial f}{\partial y} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) = \frac{\Delta \frac{\partial f}{\partial y'} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x}.$$

Derivadas parciais

$$\begin{aligned} \frac{\partial f}{\partial y} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) &= 0 \\ \frac{\partial f}{\partial y'} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) &= \frac{\frac{\Delta y_{i-1}}{\Delta x}}{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x} \right)^2}}. \end{aligned}$$

Assim,

$$\begin{aligned} \frac{\Delta \frac{\partial f}{\partial y'} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x} &= 0 \\ \Leftrightarrow \Delta \frac{\partial f}{\partial y'} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) &= 0 \\ \Leftrightarrow \frac{\partial f}{\partial y'} \left(x_{i+1}, y_{i+1}, \frac{\Delta y_i}{\Delta x} \right) - \frac{\partial f}{\partial y'} \left(x_i, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) &= 0. \end{aligned}$$

Substituindo f , obtém-se

$$\frac{\frac{y_{i+1} - y_i}{2}}{\sqrt{1 + \left(\frac{y_{i+1} - y_i}{2} \right)^2}} - \frac{\frac{y_i - y_{i-1}}{2}}{\sqrt{1 + \left(\frac{y_i - y_{i-1}}{2} \right)^2}} = 0$$

que coincide com a equação (1.8).

A partir deste ponto, todos os cálculos são iguais aos efectuados na Discretização de Guseinov.

d) Solução Óptima

A curva de comprimento mínimo que liga os pontos $(-1, 2)$ e $(5, 4)$ é a recta $y = \frac{x}{3} + \frac{7}{3}$, ou seja, a solução exacta para este problema (1.7) é a função

$$\hat{y}(x) = \frac{x}{3} + \frac{7}{3}.$$

O valor óptimo é

$$I[\hat{y}] = \int_{-1}^5 \sqrt{1 + \left(\frac{1}{3} \right)^2} dx = \frac{\sqrt{10}}{3} [5 - (-1)] = 2\sqrt{10}.$$

Neste exemplo, obteve-se a solução óptima, tanto com o método proposto por Guseinov, como com o método das Diferenças Finitas de Euler.

Exemplo 2. Determine-se uma curva, cuja revolução em torno do eixo dos x 's define uma superfície de área mínima:

$$\min_y \int_{-2}^1 y(x) \sqrt{1 + y'(x)^2} dx \quad (1.9)$$

sob as condições de fronteira

$$y(-2) = 3 \text{ e } y(1) = 2. \quad (1.10)$$

Neste caso, $f(x, y, y') = y\sqrt{1 + y'^2}$.

Discretizando o intervalo $[-2, 1]$ com 3 pontos ($n = 2$ e $\Delta x = \frac{3}{2}$), tem-se $X = \{-2, -\frac{1}{2}, 1\}$.

O problema será abordado usando a discretização de Guseinov e a discretização usual. Observar-se-á que a discretização de Guseinov conduz a melhores resultados.

a) Discretização de Guseinov

Determine-se a função auxiliar L .

Como a curva é aproximada por segmentos de recta $[(x_{i-1}, y_{i-1}), (x_i, y_i)]$, a superfície de revolução é aproximada por superfícies de troncos de cones (ou cones truncados) cuja geratriz é o segmento de recta (com comprimento $\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} = \sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}$), a altura é $x_i - x_{i-1}$ e os raios das bases são y_{i-1} e y_i .

A área da superfície do cone truncado é

$$\pi(y_{i-1} + y_i)\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2} = \pi(y_{i-1} + y_i)\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \Delta x_{i-1}.$$

Somando as áreas das superfícies dos n cones obtém-se a expressão

$$\pi \sum_{i=1}^n (y_{i-1} + y_i) \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \Delta x_{i-1},$$

ou seja, procuram-se os valores y_i que minimizem

$$J[y] = \sum_{i=1}^n (y_{i-1} + y_i) \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \Delta x_{i-1}.$$

Desta forma,

$$L(s, t, u, v, w) = (u + v)\sqrt{1 + w^2}.$$

Cálculos directos dão as derivadas parciais de L :

$$\begin{aligned} L_u &= \frac{\partial L}{\partial u}(s, t, u, v, w) = \sqrt{1 + w^2} \\ L_v &= \frac{\partial L}{\partial v}(s, t, u, v, w) = \sqrt{1 + w^2} \\ L_w &= \frac{\partial L}{\partial w}(s, t, u, v, w) = \frac{(u + v)w}{\sqrt{1 + w^2}}. \end{aligned}$$

Como se considerou, mais uma vez, $\Delta x_i = \Delta x = \frac{3}{2}$ constante, a equação de Euler-Lagrange

$$\begin{aligned} \frac{\partial L}{\partial u} \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{\Delta y_i}{\Delta x} \right) + \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) \\ = \frac{\Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x} \end{aligned}$$

é facilmente simplificada.

Considerando apenas 3 pontos, como y_0 e y_2 estão fixos ($y_0 = 3, y_2 = 2$, segundo (1.10)), haverá, no sistema de equações não lineares, apenas uma equação de Euler-Lagrange ($i = 1$):

$$\begin{aligned}
& \frac{\partial L}{\partial u} \left(-\frac{1}{2}, 1, y \left(-\frac{1}{2} \right), y(1), \frac{y(1)-y(-\frac{1}{2})}{\frac{3}{2}} \right) \\
& + \frac{\partial L}{\partial v} \left(-2, -\frac{1}{2}, y(-2), y \left(-\frac{1}{2} \right), \frac{y(-\frac{1}{2})-y(-2)}{\frac{3}{2}} \right) \\
& = \frac{\frac{\partial f}{\partial w} \left(-\frac{1}{2}, 1, y \left(-\frac{1}{2} \right), y(1), \frac{y(1)-y(-\frac{1}{2})}{\frac{3}{2}} \right) - \frac{\partial f}{\partial w} \left(-2, -\frac{1}{2}, y(-2), y \left(-\frac{1}{2} \right), \frac{y(-\frac{1}{2})-y(-2)}{\frac{3}{2}} \right)}{\frac{3}{2}} \\
& \Leftrightarrow \sqrt{1 + \left(\frac{y(1)-y(-\frac{1}{2})}{\frac{3}{2}} \right)^2} - \sqrt{1 + \left(\frac{y(-\frac{1}{2})-y(-2)}{\frac{3}{2}} \right)^2} \\
& = \frac{2}{3} \left[\frac{(y(-\frac{1}{2})+y(1)) \frac{y(1)-y(-\frac{1}{2})}{\frac{3}{2}}}{\sqrt{1 + \left(\frac{y(1)-y(-\frac{1}{2})}{\frac{3}{2}} \right)^2}} - \frac{(y(-2)+y(-\frac{1}{2})) \frac{y(-\frac{1}{2})-y(-2)}{\frac{3}{2}}}{\sqrt{1 + \left(\frac{y(-\frac{1}{2})-y(-2)}{\frac{3}{2}} \right)^2}} \right] \\
& \Leftrightarrow \sqrt{1 + \left(\frac{2(y(1)-y(-\frac{1}{2}))}{3} \right)^2} - \sqrt{1 + \left(\frac{2(y(-\frac{1}{2})-y(-2))}{3} \right)^2} \\
& = \frac{4}{9} \left[\frac{(y(-\frac{1}{2})+y(1))(y(1)-y(-\frac{1}{2}))}{\sqrt{1 + \left(\frac{2(y(1)-y(-\frac{1}{2}))}{3} \right)^2}} - \frac{(y(-2)+y(-\frac{1}{2}))(y(-\frac{1}{2})-y(-2))}{\sqrt{1 + \left(\frac{2(y(-\frac{1}{2})-y(-2))}{3} \right)^2}} \right] \\
& \Leftrightarrow \sqrt{1 + \left(\frac{2(y(1)-y(-\frac{1}{2}))}{3} \right)^2} - \sqrt{1 + \left(\frac{2(y(-\frac{1}{2})-y(-2))}{3} \right)^2} \\
& = \frac{4}{9} \left[\frac{y(1)^2 - y(-\frac{1}{2})^2}{\sqrt{1 + \left(\frac{2(y(1)-y(-\frac{1}{2}))}{3} \right)^2}} - \frac{y(-\frac{1}{2})^2 - y(-2)^2}{\sqrt{1 + \left(\frac{2(y(-\frac{1}{2})-y(-2))}{3} \right)^2}} \right].
\end{aligned}$$

Tem-se então o seguinte sistema de equações não lineares nas incógnitas $y(-2), y(-\frac{1}{2})$

e $y(1)$:

$$\left\{ \begin{array}{l} \sqrt{1 + \left(\frac{2(y(1) - y(-\frac{1}{2}))}{3} \right)^2} - \sqrt{1 + \left(\frac{2(y(-\frac{1}{2}) - y(-2))}{3} \right)^2} \\ - \frac{4}{9} \left[\frac{y(1)^2 - y(-\frac{1}{2})^2}{\sqrt{1 + \left(\frac{2(y(1) - y(-\frac{1}{2}))}{3} \right)^2}} - \frac{y(-\frac{1}{2})^2 - y(-2)^2}{\sqrt{1 + \left(\frac{2(y(-\frac{1}{2}) - y(-2))}{3} \right)^2}} \right] = 0 \\ y(-2) = 3 \\ y(1) = 2 \end{array} \right.$$

Resolvendo numericamente o sistema obtêm-se duas soluções (aproximadas)

$$\left\{ \begin{array}{l} y(-2) = 3 \\ y(-\frac{1}{2}) = 1.82617 \text{ ou } y(-\frac{1}{2}) = 0.627641 \\ y(1) = 2 \end{array} \right.$$

Com cada uma destas soluções pode definir-se uma função por ramos lineares.

Para a primeira solução ($y(-\frac{1}{2}) = 1.82617$) tem-se

$$\bar{y}_1(x) = \begin{cases} 1.43489 - 0.782553x & \text{se } -2 \leq x < -\frac{1}{2} \\ 1.88411 + 0.115887x & \text{se } -\frac{1}{2} \leq x \leq 1 \end{cases} \quad (1.11)$$

que tem como valor do funcional

$$\begin{aligned} & \sum_{i=1}^2 (y_{i-1} + y_i) \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x} \right)^2} \Delta x \\ &= \frac{3}{2} \left((y_0 + y_1) \sqrt{1 + \left(\frac{2(y_1 - y_0)}{3} \right)^2} + (y_1 + y_2) \sqrt{1 + \left(\frac{2(y_2 - y_1)}{3} \right)^2} \right) \\ &= \frac{3}{2} \left((3 + 1.82617) \sqrt{1 + \left(\frac{2(1.82617 - 3)}{3} \right)^2} + (1.82617 + 2) \sqrt{1 + \left(\frac{2(2 - 1.82617)}{3} \right)^2} \right) \\ &= 14.9701 \end{aligned}$$

Quanto à segunda solução, em que $y(-\frac{1}{2}) = 0.627641$, tem-se

$$\bar{y}_2(x) = \begin{cases} -0.163146 - 1.58157x & \text{se } -2 \leq x < -\frac{1}{2} \\ 1.08509 + 0.914906x & \text{se } -\frac{1}{2} \leq x < 1 \end{cases}$$

que tem como valor do funcional

$$\begin{aligned} & \frac{3}{2} \left((3 + 0.627641) \sqrt{1 + \left(\frac{2(0.627641 - 3)}{3} \right)^2} + (0.627641 + 2) \sqrt{1 + \left(\frac{2(2 - 0.627641)}{3} \right)^2} \right) \\ &= 15.5242. \end{aligned}$$

Seleciona-se, assim, a primeira (a que conduz a um menor valor do funcional).

b) Discretização Usual

Neste caso, a função L é determinada directamente a partir do Lagrangiano $f(x, y, y') = y\sqrt{1 + y'^2}$, ou seja,

$$L(s, t, u, v, w) = v\sqrt{1 + w^2} = f(t, v, w).$$

Tal como no caso anterior, faz-se

$$\Delta x = \frac{3}{2},$$

$$X = \left\{-2, -\frac{1}{2}, 1\right\}.$$

As derivadas parciais de L são

$$\begin{aligned} L_u &= \frac{\partial L}{\partial u}(s, t, u, v, w) = 0 \\ L_v &= \frac{\partial L}{\partial v}(s, t, u, v, w) = \sqrt{1 + w^2} \\ L_w &= \frac{\partial L}{\partial w}(s, t, u, v, w) = \frac{vw}{\sqrt{1 + w^2}}. \end{aligned}$$

e a equação de Euler-Lagrange é dada por

$$\begin{aligned} \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) &= \frac{\Delta \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right)}{\Delta x} \\ \Leftrightarrow \frac{\partial L}{\partial v} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) \Delta x &= \frac{\partial L}{\partial w} \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{\Delta y_i}{\Delta x} \right) \\ &\quad - \frac{\partial L}{\partial w} \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{\Delta y_{i-1}}{\Delta x} \right) \end{aligned}$$

que toma a forma (tem-se apenas uma equação de Euler-Lagrange ($i = 1$))

$$\begin{aligned}
& \frac{\partial L}{\partial v} \left(-2, -\frac{1}{2}, y(-2), y \left(-\frac{1}{2} \right), \frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right) \frac{3}{2} \\
&= \frac{\partial L}{\partial w} \left(-\frac{1}{2}, 1, y \left(-\frac{1}{2} \right), y(1), \frac{y(1) - y \left(-\frac{1}{2} \right)}{\frac{3}{2}} \right) \\
&\quad - \frac{\partial L}{\partial w} \left(-2, -\frac{1}{2}, y(-2), y \left(-\frac{1}{2} \right), \frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right) \\
&\Leftrightarrow \frac{3}{2} \sqrt{1 + \left(\frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right)^2} \\
&= \frac{y(1) \frac{y(1) - y \left(-\frac{1}{2} \right)}{\frac{3}{2}}}{\sqrt{1 + \left(\frac{y(1) - y \left(-\frac{1}{2} \right)}{\frac{3}{2}} \right)^2}} - \frac{y \left(-\frac{1}{2} \right) \frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}}}{\sqrt{1 + \left(\frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right)^2}} \\
&\Leftrightarrow \frac{3}{2} \sqrt{1 + \left(\frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right)^2} \\
&= \frac{2y(1) \left(y(1) - y \left(-\frac{1}{2} \right) \right)}{3 \sqrt{1 + \left(\frac{y(1) - y \left(-\frac{1}{2} \right)}{\frac{3}{2}} \right)^2}} + \frac{2y \left(-\frac{1}{2} \right) \left(y \left(-\frac{1}{2} \right) - y(-2) \right)}{3 \sqrt{1 + \left(\frac{y \left(-\frac{1}{2} \right) - y(-2)}{\frac{3}{2}} \right)^2}}.
\end{aligned}$$

Tem-se então o seguinte sistema de equações não lineares cujas incógnitas são $y(-2)$, $y \left(-\frac{1}{2} \right)$ e $y(1)$:

$$\begin{cases} \frac{9}{4} \sqrt{1 + \left(\frac{2(y \left(-\frac{1}{2} \right) - y(-2))}{3} \right)^2} = \frac{y(1)(y(1) - y \left(-\frac{1}{2} \right))}{\sqrt{1 + \left(\frac{2(y(1) - y \left(-\frac{1}{2} \right))}{3} \right)^2}} + \frac{y \left(-\frac{1}{2} \right) (y \left(-\frac{1}{2} \right) - y(-2))}{\sqrt{1 + \left(\frac{2(y \left(-\frac{1}{2} \right) - y(-2))}{3} \right)^2}} \\ y(-2) = 3 \\ y(1) = 2 \end{cases}$$

Resolvendo numericamente o sistema obtêm-se três soluções (aproximadas), sendo duas complexas:

$$\begin{cases} y(-2) = 3 \\ y \left(-\frac{1}{2} \right) = 1.98277 + 1.73884i \text{ ou } y \left(-\frac{1}{2} \right) = 1.98277 - 1.73884i \text{ ou } y \left(-\frac{1}{2} \right) = 5.59279 \\ y(1) = 2 \end{cases}$$

Apenas interessa a solução real, ou seja,

$$\begin{cases} y(-2) = 3 \\ y \left(-\frac{1}{2} \right) = 5.59279 \\ y(1) = 2 \end{cases}$$

A partir desta solução, pode definir-se uma função por ramos

$$\bar{y}_1(x) = \begin{cases} 6.45705 + 1.72853x & \text{se } -2 \leq x < -\frac{1}{2} \\ 4.39519 - 2.39519x & \text{se } -\frac{1}{2} \leq x \leq 1 \end{cases}. \quad (1.12)$$

que dá uma aproximação à solução do problema contínuo (1.9)-(1.10).

O correspondente valor do funcional discreto é

$$\begin{aligned} & \sum_{i=1}^2 y_i \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x} \right)^2} \Delta x \\ &= \frac{3}{2} \left(y_1 \sqrt{1 + \left(\frac{2(y_1 - y_0)}{3} \right)^2} + y_2 \sqrt{1 + \left(\frac{2(y_2 - y_1)}{3} \right)^2} \right) \\ &= \frac{3}{2} \left(5.59279 \sqrt{1 + \left(\frac{2(5.59279 - 3)}{3} \right)^2} + 2 \sqrt{1 + \left(\frac{2(2 - 5.59279)}{3} \right)^2} \right) \\ &= 24.5395 \end{aligned}$$

c) Solução Óptima

Para resolver este problema na sua forma contínua procuram-se soluções que verifiquem a equação (1.3) (equação de Euler-Lagrange).

Determinam-se as derivadas parciais de $f(x, y, y') = y\sqrt{1 + y'^2}$:

$$\frac{\partial f}{\partial y}(x, y, y') = \sqrt{1 + y'^2}$$

$$\frac{\partial f}{\partial y'}(x, y, y') = \frac{y \cdot y'}{\sqrt{1 + y'^2}}$$

$$\frac{d}{dx} \left(\frac{\partial f}{\partial y'}(x, y, y') \right) = \frac{y'^2}{\sqrt{1 + y'^2}} + \frac{y \cdot y'}{\sqrt{1 + y'^2}} - \frac{y \cdot y'^2 \cdot y''}{(1 + y'^2)^{\frac{3}{2}}}.$$

A equação diferencial (1.3) tem a forma:

$$\sqrt{1 + y'^2} = \frac{y'^2}{\sqrt{1 + y'^2}} + \frac{y \cdot y'}{\sqrt{1 + y'^2}} - \frac{y \cdot y'^2 \cdot y''}{(1 + y'^2)^{\frac{3}{2}}}.$$

Resolvendo esta equação tendo em conta as condições de fronteira do problema (1.10) obtém-se o extremal³:

$$\hat{y}(x) = 0.8435613585 \left(\frac{1}{(e^{0.5927251112x})^2 (e^{0.007251334202} + 1)} \right) e^{0.5927251112x} e^{0.007251334202}. \quad (1.13)$$

O valor óptimo é

$$I[\hat{y}] = \int_{-2}^1 \hat{y}(x) \sqrt{1 + \hat{y}'(x)^2} dx = 7.430431404.$$

³Cálculos efectuados recorrendo à ferramenta *Maple*, encontrando-se os comandos utilizados em anexo (§A.1, página 79).

Observando os valores dos funcionais discretos, constata-se que a discretização de Guseinov encontra uma solução melhor do que a discretização usual. No entanto, comparando estes valores (14.9701 e 24.5395) com o valor ótimo (7.4304), as aproximações não parecem razoáveis.

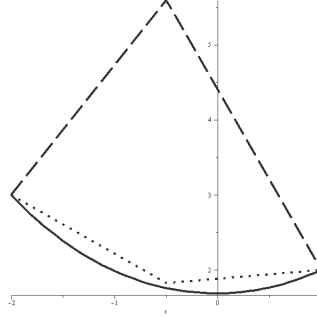


Figura 1.1: Gráfico que mostra as três curvas (1.13) (linha), (1.11) (pontilhada) e (1.12) (tracejada).

Tendo em conta que, em grande parte dos problemas do Cálculo das Variações se procuram curvas e as diferentes discretizações encontram pontos que permitem definir curvas, calcule-se o valor do funcional (contínuo) para cada uma destas curvas (ver Figura 1.1).

Com a função definida por ramos lineares (1.11), obtida pela discretização de Guseinov, tem-se 7.48503 como valor do integral.

O valor do integral calculado com a função (1.12), também definida por ramos lineares e determinada através da discretização usual é 27.6502.

Estes novos dados vêm corroborar que, para este exemplo, a discretização de Guseinov apresenta melhores soluções do que a discretização usual. Neste exemplo, a solução aproximada encontrada através da discretização usual aplicada às equações de Euler-Lagrange aqui propostas parece não fazer sentido.

Além disso, considerando apenas 3 pontos, a solução proposta, usando a discretização de Guseinov, parece ser razoável. Espera-se que com mais pontos se obtenham melhores soluções à custa de maior esforço computacional.

1.3 Conclusão

Neste capítulo foram apresentadas as estratégias de Euler e de Lagrange para problemas do Cálculo das Variações.

Discutiram-se diferentes abordagens discretas para estes problemas, dando ênfase ao método de Guseinov ([4]).

Os resultados apresentados neste capítulo são fundamentais para explicar o método implementado neste trabalho e que será usada para obtenção de soluções aproximadas de vários problemas do Cálculo das Variações.

Capítulo 2

Formulações e Ferramentas Computacionais

2.1 Introdução

Existe uma grande variedade de áreas em Optimização, cada uma com diferentes métodos e técnicas para resolver os respectivos problemas. Vão também surgindo novos problemas mas também novas técnicas para resolver problemas menos recentes. Além disso, é cada vez mais fácil implementar os métodos em diferentes linguagens de programação ou ferramentas computacionais com muitas das funções matemáticas necessárias já incorporadas.

Por outro lado, é muitas vezes possível converter problemas de determinada forma em problemas com outra formulação, permitindo encontrar soluções para o mesmo problema de diferentes modos.

Neste capítulo são apresentadas algumas ferramentas computacionais e métodos que podem ser usados para resolver problemas do Cálculo das Variações.

Foram encontradas poucas ferramentas específicas para resolver problemas do Cálculo das Variações. Além de ferramentas desenvolvidas em ambiente lectivo, existe, por exemplo, a *VariationalCalculus* do *Maple* 10, uma ferramenta que permite resolver analiticamente problemas do Cálculo das Variações¹: determinar as equações de Euler-Lagrange a partir do funcional (contínuo) do problema e resolver a equação diferencial obtida.

No entanto, existem ferramentas que resolvem problemas de Programação Não Linear e de Controlo Óptimo. Como já se referiu, é possível converter problemas de uma determinada forma em problemas de outra forma. Por isso, antes de apresentar as ferramentas e os métodos usados para resolver alguns exemplos de problemas do Cálculo das Variações, vai mostrar-se como é feita a conversão de um problema do Cálculo das Variações num problema de Controlo Óptimo ou num problema de Programação Não Linear.

¹No Anexo A.2 (página 81) encontra-se o modo de aplicação desta ferramenta para resolver o problema do Exemplo 1 apresentado no Capítulo 1 (página 12).

2.2 Diferentes formulações do problema

Já se viu que num problema do Cálculo das Variações se pretende encontrar funções continuamente diferenciáveis y que minimizem um dado funcional

$$\int_{x_a}^{x_b} f(x, y(x), y'(x)) dx$$

tal que $y(x_a) = \alpha$ e $y(x_b) = \beta$, sendo $x_a, x_b, \alpha, \beta \in \mathbb{R}$ com $x_a < x_b$ e $f : [x_a, x_b] \times \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ uma função contínua e diferenciável com continuidade.

Cálculo das Variações como caso particular do Controlo Óptimo

O Controlo Óptimo surgiu (como área da optimização) no final da década de 50 do século XX na antiga União Soviética e é uma extensão do Cálculo das Variações.

Um problema de Controlo Óptimo tem:

- Variável(eis) de Estado do Sistema (a função y);
- Variável(eis) de Controlo do Sistema, denotada(s) por z (tal que $\forall x \in [x_a, x_b] \ z(x) \in Z$, com Z um dado conjunto);
- Equação ou Sistema de Controlo (caso haja apenas uma ou várias equações, respectivamente): $y'(x) = \frac{dy}{dx}(x) = \varphi(x, y(x), z(x))$;
- Condições de Fronteira (tal como no Cálculo das Variações): $y(x_a) = \alpha$ e $y(x_b) = \beta$, com $x_a < x_b \in \mathbb{R}$ e $\alpha, \beta \in \mathbb{R}$, em que x_b e β podem estar fixos ou não.

Observação 2.2.1. *As funções y e z são aqui consideradas reais de variável real mas podem ser funções vectoriais de variável real ($y, z : \mathbb{R} \rightarrow \mathbb{R}^n$ e $\alpha, \beta \in \mathbb{R}^n$). No entanto, para os exemplos estudados, basta que sejam funções escalares ($n = 1$).*

Os problemas de Controlo Óptimo podem ter restrições aos valores das variáveis de controlo ($Z \subset \mathbb{R}$), que não existem no Cálculo das Variações (no Cálculo das Variações tem-se sempre $Z = \mathbb{R}$). Desta forma, há problemas de Controlo Óptimo que não podem ser resolvidos (nem sequer formulados) usando o Cálculo das Variações.

Por outro lado, muitas vezes os problemas de Controlo Óptimo são apresentados de forma diferente, principalmente o funcional que se pretende extremar. A escolha entre as diferentes formulações pode ter a ver com o problema em estudo ou com o método a usar para resolver

o problema.

O Princípio do Máximo de Pontryagin é uma condição necessária de optimalidade do Controlo Ótimo que generaliza as equações de Euler-Lagrange (usadas para encontrar soluções para problemas do Cálculo das Variações). Este resultado é usado para encontrar soluções para problemas de Controlo Ótimo tendo, também, formulações ajustadas à colocação do problema.

Abordagem Lagrangiana

Seja $L : [x_a, x_b] \times \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ uma função contínua e diferenciável com continuidade. Pretende-se encontrar o par de funções $(y(\cdot), z(\cdot))$ que minimiza

$$\int_{x_a}^{x_b} L(x, y(x), z(x)) dx \quad (2.1)$$

$$s.a \ y'(x) = \varphi(x, y(x), z(x)). \quad (2.2)$$

Com esta formulação - *formulação de Lagrange* - basta considerar $\varphi(x, y(x), z(x)) = z(x)$ ($y'(x) = z(x)$) para que se tenha um problema do Cálculo das Variações na forma de Lagrange do Controlo Ótimo.

Abordagem Hamiltoniana

Para resolver um problema de Controlo Ótimo usando o *Princípio do Máximo de Pontryagin* (Teorema 2.2.2) faz-se um ajuste semelhante ao que é utilizado nos problemas de Programação Matemática ao aplicar o *Método dos Multiplicadores de Lagrange* ([10],[12]). Para isto deve considerar-se uma constante ψ_0 (associada ao funcional a extremar), uma função real de variável real ψ (associada à restrição do problema) e o Hamiltoniano definido por

$$H(x, y, z, \psi_0, \psi) = \psi_0 L(x, y, z) + \psi \varphi(x, y, z).$$

Como uma das condições do problema é $\varphi(x, y(x), z(x)) = y'(x)$ (sistema de controlo) e o *Método dos Multiplicadores de Lagrange* considera que as restrições são expressões igualadas a zero ($\varphi(x, y(x), z(x)) - y'(x) = 0$), nesta abordagem ao problema pretende extremar-se

$$\begin{aligned} & \int_{x_a}^{x_b} [\psi_0 L(x, y(x), z(x)) + \psi(x) (\varphi(x, y(x), z(x)) - y'(x))] dx \\ &= \int_{x_a}^{x_b} [H(x, y(x), z(x), \psi_0, \psi(x)) - \psi(x) \cdot y'(x)] dx. \end{aligned}$$

Teorema 2.2.2 (Princípio do Máximo de Pontryagin). *Se o par $(y(\cdot), z(\cdot))$ ($z(x) \in Z \subseteq \mathbb{R}$) minimiza o problema de Controlo Ótimo (2.1)–(2.2) então existe um par não nulo $(\psi_0, \psi(\cdot))$, onde $\psi_0 \leq 0$ é uma constante e ψ é uma função contínua de domínio $[x_a, x_b]$ com a primeira derivada contínua ($\psi \in C^1$) e com valores em \mathbb{R} tal que as seguintes condições são válidas para qualquer $x \in [x_a, x_b]$:*

1. Sistema Hamiltoniano

$$\begin{cases} y'(x) &= \frac{\partial H}{\partial \psi}(x, y(x), z(x), \psi_0, \psi(x)) \\ \psi'(x) &= -\frac{\partial H}{\partial y}(x, y(x), z(x), \psi_0, \psi(x)) \end{cases} ;$$

2. Condição de Máximo

$$H(x, y(x), z(x), \psi_0, \psi(x)) = \max_{\hat{z} \in Z} (x, y(x), \hat{z}, \psi_0, \psi(x)) .$$

Demonstração. Ver [10]. □

Observação 2.2.3. *Tal como referido anteriormente, consideram-se escalares as funções (y, z, ψ) , apesar do Princípio Máximo de Pontryagin ser válido para funções vectoriais.*

Chama-se *extremal de Potryagin* ao quadruplo $(y(\cdot), z(\cdot), \psi_0, \psi(\cdot))$ que verifica as condições do Teorema 2.2.2. Além disso, se $\psi_0 < 0$, a extremal diz-se *normal*. Caso ψ_0 seja nulo, diz-se que a extremal é *anormal*.

A primeira equação no Sistema Hamiltoniano é, nada mais que, o *Sistema de Controlo* e a segunda $(\psi' = -\frac{\partial H}{\partial y})$ chama-se *Sistema Adjunto*.

Problema na forma de Bolza

Os funcionais do Cálculo das Variações e do Controlo Óptimo foram apresentadas até aqui na forma de Lagrange. Outra formulação dos problemas consiste em considerar funcionais na forma de Bolza:

$$\mathcal{L}(x_a, \alpha, x_b, \beta) + \int_{x_a}^{x_b} L(x, y(x), z(x)) dx. \quad (2.3)$$

As restantes condições mantêm-se: $y'(x) = \varphi(x, y(x), z(x))$, $y(x_a) = \alpha$ e $y(x_b) = \beta$. Os argumentos da nova função \mathcal{L} são os valores das condições de fronteira. Além disso, x_a e x_b podem, ou não, ser fixos.

A forma de Lagrange é um caso particular desta formulação de Bolza ($\mathcal{L} \equiv 0$).

Problema na forma de Mayer

Existe ainda a formulação dos problemas na forma de Mayer (outro caso particular do *problema de Bolza*, considerando $L \equiv 0$) que procura extremar

$$\mathcal{L}(x_a, \alpha, x_b, \beta,) . \quad (2.4)$$

Resta ainda averiguar como se reescreve um problema dado na formulação de Bolza (por exemplo), como um problema de Controlo Óptimo (ou do Cálculo das Variações) na forma de Mayer ou de Lagrange (as diferentes formulações são, na verdade, equivalentes).

Para este fim parte-se de uma representação mais geral da formulação de Bolza que inclui outras restrições como as existentes nos problemas isoperimétricos²:

$$\begin{aligned} F_0(x_a, x_b, y(x), z(x)) &\rightarrow \min \\ y'(x) &= \varphi(x, y(x), z(x)) \\ F_i(x_a, x_b, y(x), z(x)) &\leq 0, \quad i = 1, \dots, m \\ F_i(x_a, x_b, y(x), z(x)) &= 0, \quad i = m+1, \dots, m+l \end{aligned}$$

sendo

$$F_i(x_a, x_b, y(x), z(x)) = \mathcal{L}_i(x_a, \alpha, x_b, \beta) + \int_{x_a}^{x_b} L_i(x, y(x), z(x)) dx, \quad i = 0, 1, \dots, m+l.$$

Tendo em conta este problema, consideram-se novas variáveis (funções) u_i associadas a cada L_i :

$$\begin{aligned} \frac{d}{dx} u_i(x) &= L_i(x, y(x), z(x)) \\ u_i(x_a) &= 0 \end{aligned} \quad i = 0, 1, \dots, m+l.$$

Assim, para qualquer $i = 0, 1, \dots, m+l$,

$$F_i(x_a, x_b, y(x), z(x)) = \mathcal{L}_i(x_a, \alpha, x_b, \beta) + (u_i(x_b) - u_i(x_a)) = \mathcal{L}_i(x_a, \alpha, x_b, \beta) + u_i(x_b).$$

A formulação (de Mayer) do problema é então

$$\begin{aligned} \mathcal{L}_0(x_a, \alpha, x_b, \beta) + u_0(x_b) &\rightarrow \min \\ y'(x) &= \varphi(x, y(x), z(x)) \\ u'_i(x) &= L_i(x, y(x), z(x)), \quad i = 0, \dots, m+l \\ u_i(x) &\in U, \quad i = 0, \dots, m+l, \\ U &= \{u : [x_a, x_b] \rightarrow \mathbb{R} \text{ tal que } u(x_a) = 0\} \\ \mathcal{L}_i(x_a, x_b, y(x), z(x)) + u_i(x_b) &\leq 0, \quad i = 1, \dots, m \\ \mathcal{L}_i(x_a, x_b, y(x), z(x)) + u_i(x_b) &= 0, \quad i = m+1, \dots, m+l \end{aligned}$$

²Considerando $x_a, x_b, \alpha, \beta \in \mathbb{R}$ com $x_a < x_b$ e $f : [x_a, x_b] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ uma função de classe C^1 , no problema isoperimétrico do Cálculo das Variações procura-se uma função y que minimize ou maximize

$$\int_{x_a}^{x_b} f(x, y(x), y'(x)) dx$$

de modo que, além das condições de fronteira

$$y(x_a) = \alpha \text{ e } y(x_b) = \beta,$$

seja verificada a condição isoperimétrica

$$\int_{x_a}^{x_b} g(x, y(x), y'(x)) dx = C, \text{ para um dado valor real } C.$$

Para obter um funcional de Lagrange basta

$$\begin{aligned}
F_i(x_a, x_b, y(x), z(x)) &= \mathcal{L}_i(x_a, \alpha, x_b, \beta) + \int_{x_a}^{x_b} L_i(x, y(x), z(x)) dx \\
&= \int_{x_a}^{x_b} \frac{\mathcal{L}_i(x_a, \alpha, x_b, \beta)}{x_b - x_a} dx + \int_{x_a}^{x_b} L_i(x, y(x), z(x)) dx \\
&= \int_{x_a}^{x_b} \left(\frac{\mathcal{L}_i(x_a, \alpha, x_b, \beta)}{x_b - x_a} + L_i(x, y(x), z(x)) \right) dx, \quad i = 0, 1, \dots, m + l.
\end{aligned}$$

Ou seja, considera-se um novo Lagrangiano

$$W_i(x, y(x), z(x)) = \frac{\mathcal{L}_i(x_a, \alpha, x_b, \beta)}{x_b - x_a} + L_i(x, y(x), z(x)), \quad i = 0, 1, \dots, m + l.$$

Cálculo das Variações via Programação Não Linear

Há uma diferença fundamental entre problemas do Cálculo das Variações e problemas de Programação Não Linear. Como já se viu, no primeiro caso, procura-se uma função que maximize ou minimize um funcional enquanto que no caso dos problemas de Programação Não Linear, o objectivo é encontrar um ponto (ou um conjunto de pontos) que maximize ou minimize uma função.

Os problemas de Programação Não Linear são compostos por:

- uma função objectivo: expressão não linear que se pretende extremar (maximizar ou minimizar)

$$f(x);$$

- um conjunto de restrições: equações ou inequações, lineares ou não lineares, que devem ser verificadas pela solução

$$\begin{aligned}
g_i(x) &\leq 0, \quad i = 1, 2, \dots, p \\
h_j(x) &= 0, \quad j = 1, 2, \dots, q.
\end{aligned}$$

Formulação:

$$\begin{aligned}
&\text{extr} \quad f(x) \\
&\text{s.a} \quad g_i(x) \leq 0 \\
&\quad \quad h_j(x) = 0
\end{aligned}$$

em que

$$\begin{aligned}
&x \in \mathbb{R}^n \\
&f, g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}, \text{ com } i = 1, 2, \dots, p \text{ e } j = 1, 2, \dots, q \\
&n, p, q \in \mathbb{N}.
\end{aligned}$$

Também os problemas do Cálculo das Variações podem ser formulados explicitando a expressão que se pretende extremar e as restrições a que está sujeita a solução para o problema em causa:

$$\begin{aligned} \min \quad & \int_{x_a}^{x_b} L(x, y(x), z(x)) dx \\ \text{s.a} \quad & y'(x) = z(x) \\ & y(x_a) = \alpha \\ & y(x_b) = \beta \end{aligned}$$

Estes problemas podem ser resolvidos usando, por exemplo, o já referido *Método dos Multiplicadores de Lagrange* (ver [10], pp. 39–44).

Neste caso, não se obtém um problema de Programação Não Linear convencional (tendo em conta o tipo de soluções procuradas) mas é usado um método para resolução deste tipo de problemas.

2.3 Ferramentas Computacionais

OPTRAGEN - ferramenta do Matlab

OPTRAGEN é uma ferramenta do Matlab desenvolvida por Raktim Bhattacharya para gerar trajectórias óptimas (OPTimal TRAjectory GENeration) ([2]). Esta ferramenta converte problemas de Controlo Óptimo em problemas de Programação Não Linear (através da parametrização de trajectórias como *splines*) tornando possível a resolução numérica destes problemas.

Dado um problema de Controlo Óptimo na formulação de Bolza

$$\begin{aligned} \min \quad & L(x_a, \alpha, z(x_a)) + L(x_b, \beta, z(x_b)) + \int_{x_a}^{x_b} L(x, y(x), z(x)) dx \\ \text{s. a} \quad & y'(x) = \varphi(x, y(x), z(x)) \\ & l_a \leq \Psi(x_a, \alpha, z(x_a)) \leq u_a \\ & l \leq \Psi(x, y(x), z(x)) \leq u \quad (\text{restrições de } Z) \\ & l_b \leq \Psi(x_b, \beta, z(x_b)) \leq u_b \end{aligned}$$

isto é, considerando $\mathcal{L}(x_a, \alpha, x_b, \beta) = L(x_a, \alpha, z(x_a)) + L(x_b, \beta, z(x_b))$, o OPTRAGEN devolve uma função custo (função objectivo) que se pretende minimizar e uma função de restrições

não lineares:

$$\begin{aligned} \min_{\rho} \quad & F(\rho) \\ \text{s. a} \quad & L \leq \begin{bmatrix} \rho \\ A_{\rho} \\ G(\rho) \end{bmatrix} \leq U. \end{aligned}$$

São, assim, redefinidas variáveis e restrições num problema de programação não linear equivalente que pode ser resolvido por qualquer ferramenta adequada a este tipo de problemas.

As funções y e z são parametrizadas por *B-Splines*, isto é, tanto y como z são escritas como combinação linear de curvas polinomiais (extensões da curva de *Bézier*), sendo os coeficientes as novas variáveis do problema de Programação Não Linear.

Tendo em conta a formulação inicial escolhida, o OPTRAGEN funciona bem em problemas de mecânica com restrições (de movimento, por exemplo) e condição final (x_b e β) fixa ou não.

OC - Ferramenta de Controlo Óptimo

Os autores do livro *Curso de Optimização - Programação Matemática, Cálculo das Variações, Controlo Óptimo* ([10]) desenvolveram duas ferramentas: uma para problemas de Programação Matemática e outra para problemas de Controlo Óptimo.

Já se viu que é possível converter problemas do Cálculo das Variações em problemas de Controlo Óptimo. No caso da ferramenta OC, os problemas devem ter a formulação de Mayer considerando um novo parâmetro vectorial s

$$\begin{aligned} \mathcal{L}_0(y(x_b)) &\rightarrow \min \\ y'(x) &= \varphi(x, y(x), z(x), s), \quad x \in [x_a, x_b] \\ z &\in Z, s \in S \\ y(x_a) &= g(s) \\ \mathcal{L}_i^0(s) &\leq 0, \quad i = 1, 2, \dots, m_0 \\ \mathcal{L}_i^0(s) &= 0, \quad i = m_0 + 1, \dots, m_0 + l_0 \\ \mathcal{L}_j^1(y(x_b), s) &\leq 0, \quad j = 1, 2, \dots, m_1 \\ \mathcal{L}_j^1(y(x_b), s) &= 0, \quad j = m_1 + 1, \dots, m_1 + l_1. \end{aligned}$$

Os conjuntos Z e S são paralelepípedos que contêm as soluções para z e s respectivamente.

Esta ferramenta aproxima o problema de Controlo Óptimo a um problema de Programação Matemática por discretização do intervalo $[x_a, x_b]$ em n subintervalos. Usando métodos de Programação Matemática, a OC obtém soluções aproximadas para o problema, permitindo escolher alguns parâmetros. São eles:

- método de optimização:
 - Gradientes Conjugados (*Conjugated Gradients* - CG);

- Newton (*Newton Method* - NM);
- “Pesquisa Unidireccional” (*Univariate Search* - US)
- Pesquisa Directa (*Direct Search* - DS 1 e DS 2);
- Pesquisa Aleatória (*Random Search* - RS);
- método para resolver equações diferenciais:
 - Euler (E);
 - Runge-Kutta (R-K 2);
- solução inicial;
- número máximo de iterações;
- número de intervalos (da discretização).

2.4 Algoritmo evolucionista

Já vários algoritmos foram usados para encontrar soluções para problemas de Cálculo de Variações e continuam a surgir novas técnicas que o permitem fazer. Cada vez mais (e em todos os tipos de problemas), as técnicas são *híbridas*, aplicando-se diferentes áreas da matemática, algumas clássicas outras actualmente em desenvolvimento. É o que acontece com o algoritmo aqui usado para comparação - (μ, λ) -ES Simplificado ([3]).

Este algoritmo é uma simplificação de um algoritmo que usa estratégias evolucionistas combinadas com Controlo Óptimo. Vai procurando soluções para o problema, avaliando cada uma delas. As soluções mais próximas do objectivo definido são usadas para encontrar novas soluções (supostamente melhores). Este processo termina depois de um determinado número de iterações.

Input:

$$J[y(\cdot)] = \int_{x_a}^{x_b} f(x, y(x), y'(x)) dx$$

condições de fronteira: $y(x_a) = \alpha$ e $y(x_b) = \beta$

número de pontos a considerar para discretizar $[x_a, x_b]$: n

número de iterações: N

número de soluções usadas para melhorar: μ

número de soluções “melhoradas” a gerar: λ

Output: vector y de dimensão n

Algoritmo

Início

1. Definir $X \leftarrow [x_a, x_1, \dots, x_i, \dots, x_{n-2}, x_b]$ em que $x_i - x_{i-1} = \frac{x_b - x_a}{n-1}$;

2. $Y \leftarrow [y_a, y_1, \dots, y_{n-2}, y_b]$;

Gerar aleatoriamente uma função y por ramos lineares que aproxima Y

Transformar y de modo que sejam verificadas todas as condições de admissibilidade:

- fronteira - $y_a = y(x_a)$ e $y_b = y(x_b)$;
- $y'(x)$;

Se y ou y' não são admissíveis, então colocar y e y' na fronteira;

3. Repetir N vezes:

- $Y^c \leftarrow y + N(0, \sigma^2)$, $c = 1, 2, \dots, \lambda$, onde $N(0, \sigma^2)$ é um vector de perturbações aleatórias obtidas de uma distribuição normal.

Transformar cada Y^c de modo que sejam verificadas todas as condições de admissibilidade para y :

- fronteira - $y_a = y(x_a)$ e $y_b = y(x_b)$;
- $y'(x)$;

- $J^c \leftarrow J[Y^c]$, $c = 1, 2, \dots, \lambda$
 j é tal que $J^j = \min_{c=1, \dots, \lambda} J^c$
 $y \leftarrow Y^j$

Fim.

Os resultados obtidos com este algoritmo dependem de vários parâmetros : μ, n e N .

2.5 Conclusão

Neste capítulo mostrou-se que é possível formular um problema do Cálculo das Variações de diferentes formas, nomeadamente e como caso particular de um problema de Controlo Óptimo, usando a formulação de Lagrange (2.1), a formulação de Bolza (2.3) ou a formulação de Mayer (2.4). Existem, também, vários métodos e ferramentas (OC, OPTRAGEN, por exemplo) para resolver estes problemas.

Resta ainda averiguar se esses métodos e ferramentas apresentam soluções igualmente satisfatórias para o mesmo problema ou se algumas ferramentas são mais adequadas a determinado tipo de problemas do que a outros.

Além disso, não se fez qualquer referência ao modo de utilização das ferramentas. No próximo capítulo averiguar-se-á o quão imediato ou trabalhoso é o ajuste e a aplicação de cada problema a cada ferramenta, comparando depois os resultados obtidos.

Capítulo 3

Implementação e Comparação

3.1 Introdução

Há diferentes estratégias e programas para encontrar soluções para os mesmos problemas. No entanto, de um modo geral, estes métodos fornecem aproximações da solução e, por isso, olhando para cada método ou cada ferramenta isoladamente, pode não se ter a certeza do quão próxima do óptimo cada solução obtida está.

Alguns métodos podem fornecer soluções muito próximas da solução óptima mas têm elevado tempo de execução.

Neste capítulo serão comparados algoritmos e a ferramenta OC (apresentados no Capítulo 2) com o método apresentado de seguida em problemas cuja solução óptima é conhecida: Problema da Braquistócrona e Problema de Manià (que verifica o Fenómeno de Lavrentiev).

O algoritmo em estudo foi implementado no *Mathematica* 6 (versão 6.0.1.0), uma ferramenta computacional que já tem várias funções (de cálculo matemático) incorporadas que permitem, por exemplo, calcular derivadas parciais de funções, resolver sistemas de equações não lineares, algébrica e numericamente, entre outras.

Os elementos a comparar são o valor do integral do Lagrangiano da função aproximada das soluções e o modo de utilização de cada ferramenta.

3.2 Algoritmo sob investigação

Baseado no Teorema 1.2.2 apresentado no final Capítulo 1 (página 8) tem-se o seguinte algoritmo para determinar computacionalmente soluções para problemas do Cálculo das Variações.

Algoritmo para a resolução aproximada do problema fundamental do Cálculo das Variações

Input:

- funcional contínuo: f
- Lagrangiano Discreto: L
- condições de fronteira:

$$y(x_a) = \alpha \text{ e } y(x_b) = \beta$$

- número de intervalos em que se divide o domínio de integração ¹: n

Output:

- $y(x_i), i = 0, 1, \dots, n$
- $J[y]$
- $\int_{x_a}^{x_b} f(x, \bar{y}(x), \bar{y}'(x)) dx$, sendo \bar{y} a função definida por ramos lineares a partir dos pontos $y(x_i)$ encontrados

Algoritmo

1. Leitura dos dados de entrada:

$$L(s, t, u, v, w)$$

$$t_0 \leftarrow x_a$$

$$t_n \leftarrow x_b$$

$$\alpha \leftarrow y_a$$

$$\beta \leftarrow y_b$$

$$n$$

¹As equações de Euler-Lagrange encontradas no referido teorema consideram quaisquer valores para os x 's no intervalo e em determinado número mas na implementação optou-se por escolher x 's igualmente espaçados (ou seja, Δx constante).

2. Cálculos iniciais:

Determinação das derivadas parciais de L

$$Lu \leftarrow \frac{\partial L}{\partial u}(s, t, u, v, w)$$

$$Lv \leftarrow \frac{\partial L}{\partial v}(s, t, u, v, w)$$

$$Lw \leftarrow \frac{\partial L}{\partial w}(s, t, u, v, w)$$

Determinação do conjunto de $n + 1$ x 's equidistantes:

- $\Delta x = \frac{x_b - x_a}{n}$ constante;
- $X = \{x_0 = x_a, x_1 = x_0 + \Delta x, \dots, x_i = x_0 + i\Delta x, \dots, x_n = x_b\}$.

3. Resolução do sistema de equações não lineares relativo às *Equações de Euler-Lagrange* no caso discreto

- Determinação das equações do sistema:

$F(Y) = 0$, em que

– para i de 1 até $n - 1$,

$$\begin{aligned} F_i = & Lu \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) \\ & + Lv \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right) \\ & - \frac{Lw \left(x_i, x_{i+1}, y_i, y_{i+1}, \frac{y_{i+1} - y_i}{\Delta x} \right) - Lw \left(x_{i-1}, x_i, y_{i-1}, y_i, \frac{y_i - y_{i-1}}{\Delta x} \right)}{\Delta x} \end{aligned}$$

– condições de fronteira

$$F_0 = y_0 - \alpha$$

$$F_n = y_n - \beta$$

- Resolução do Sistema

4. Cálculo de $J[y(\cdot)] = \sum_{i=1}^n L \left(x_{i-1}, x_i, y(x_{i-1}), y(x_i), \frac{\Delta y_{i-1}}{\Delta x} \right) \Delta x$;

5. Determinação da função, definida por ramos lineares, \bar{y} ;

6. Cálculo de $\int_{x_a}^{x_b} f(x, \bar{y}(x), \bar{y}'(x)) dx$;

7. Representação gráfica da função \bar{y} .

Para usar esta implementação basta discretizar o problema do Cálculo das Variações e escrever o Lagrangiano Discreto (L), o Lagrangiano Contínuo(f), as Condições de Fronteira (x_a, x_b, y_a e y_b) e o número de intervalos a considerar (n) no *notebook* do *Mathematica*.

Pormenores de Implementação

Tendo em conta o problema e o número de pontos a considerar, o Sistema de Equações Não Lineares não é, geralmente, de simples resolução. Por isto, as funções (do *Mathematica*) para resolução de sistemas de equações podem não apresentar soluções (resolução simbólica) ou apresentar soluções complexas (resolução numérica).

Usa-se, então, nesta implementação, uma função que resolve numericamente problemas de Programação Não Linear, obtendo-se desta forma, uma aproximação para a solução do sistema de equações.

Sistema de Equações Não Lineares

$$\begin{cases} F(i) = 0, & i = 1, \dots, n-1 \\ y_0 - \alpha = 0 \\ y_n - \beta = 0 \end{cases}$$

Problema de Programação Não Linear

$$\begin{aligned} \min \quad & \sum_{i=1}^{n-1} (F(i))^2 + (y_0 - \alpha)^2 + (y_n - \beta)^2 \\ \text{s.a} \quad & y_0 = \alpha \\ & y_n = \beta \\ & y_i \in \mathbb{R} \end{aligned}$$

Aplicou-se a função *NMinimize*² para obter uma aproximação numérica da solução do problema não linear. É possível escolher o método a usar na procura dos pontos:

- *Random Search*

Este algoritmo procura a melhor solução de um conjunto de soluções geradas aleatoriamente. Neste caso, a melhor solução minimiza localmente a função objectivo. A esta solução (agora considerada solução inicial) é aplicado um método iterativo que deverá convergir para o mínimo local.

- *Nelder Mead*

Este algoritmo é um método simplex (para Programação Não Linear sem restrições) muito usado por efectuar a pesquisa de soluções sem calcular derivadas. Em cada iteração do método é considerado um politopo convexo com $n + 1$ vértices caso se procurem n pontos. Estes $n + 1$ pontos são um *simplex*.

²<http://reference.wolfram.com/mathematica/tutorial/ConstrainedOptimizationGlobalNumerical.html>

Determina-se o valor da função objectivo para cada um dos pontos e exclui-se o ponto que apresenta maior valor. É, então, gerada uma nova solução para substituir a excluída. Este processo continua de forma a diminuir o polítopo e, consequentemente, a região onde se encontra a solução óptima (que minimiza a função objectivo). Quando a região for pequena o suficiente, escolhe-se a melhor solução das que definem o polítopo.

- *Differential Evolution*

Como o nome indica, este algoritmo usa uma estratégia evolucionista (“mutação”). Em cada iteração, são gerados m pontos (sendo $m > n$, o número de pontos procurados) dos quais são escolhidas, aleatoriamente, m combinações de 3 pontos a usar na criação de um novo ponto. Caso o valor da função objectivo, neste novo ponto, seja melhor do que o valor da função objectivo num dos pontos que lhe deu origem, ocorre uma substituição. O processo termina convergindo quando a diferença entre os valores da função objectivo em duas soluções sucessivas é menor do que um valor previamente definido, assim como a distância entre dois “melhores” pontos de duas iterações seguidas.

- *Simulated Annealing*

Este método gera, em cada iteração, uma nova solução numa vizinhança da solução que se conhece. O raio da vizinhança diminui em cada iteração. Para cada ponto da solução, caso o novo valor encontrado seja melhor que o existente, ocorre uma substituição, caso contrário é calculado um novo valor usando estes dois valores e uma função probabilidade.

Estes métodos podem, ou não, apresentar diferentes soluções. Uma destas soluções pode ser a melhor em termos de problema de Programação Não Linear (resolução do sistema de equações não lineares) mas não o ser em relação ao problema inicial. Assim, das soluções apresentadas pelos diferentes métodos, escolhe-se aquela que minimiza o funcional do problema do Cálculo das Variações Discreto.

3.3 Exemplo 1: Problema da Braquistócrona

3.3.1 Apresentação do Problema

O **Problema da Braquistócrona** proposto por Johann Bernoulli em 1696 pode ser enunciado do seguinte modo:

Dados dois pontos $A(x_a, y_a)$ e $B(x_b, y_b)$ num mesmo plano vertical, determinar a curva descrita por uma partícula que, com velocidade inicial nula e sob acção da gravidade, una os pontos em tempo mínimo, desprezando o atrito.

$$\min_y \frac{1}{\sqrt{2g}} \int_{x_a}^{x_b} \sqrt{\frac{1 + y'(x)^2}{y_a - y(x)}} dx \quad (3.1)$$

sendo g a constante gravitacional.

A solução para este problema é um arco de cicloide, a curva descrita pela revolução de um ponto pertencente a uma circunferência de raio r que rola, sem deslizar, sobre uma recta $y = y_a$. Esta curva é descrita pela parametrização:

$$\begin{cases} x = x_a + r(\theta - \sin(\theta)) \\ y = y_a - r(1 - \cos(\theta)) \end{cases}, \quad \theta_0 \leq \theta \leq \theta_1. \quad (3.2)$$

São conhecidos dois pontos pelos quais passa a curva $((x_a, y_a)$ e $(x_b, y_b))$ sendo, por isso, possível determinar r , θ_0 e θ_1 . Usando o ponto (x_a, y_a) e considerando que $r \neq 0$ tem-se que $\theta_0 = 0$.

O tempo que a partícula demora a percorrer a curva é calculado com a expressão

$$\frac{1}{\sqrt{2g}} \int_{x_a}^{x_b} \sqrt{\frac{1 + y'(x)^2}{y_a - y(x)}} dx = \frac{1}{\sqrt{2g}} \int_0^{\theta_1} \sqrt{\frac{1 + \left(\frac{dy}{dx}(\theta)\right)^2}{y_a - y(\theta)}} \frac{dx}{d\theta}(\theta) d\theta. \quad (3.3)$$

Formulações do Problema da Braquistócrona

- Contínua

$$\begin{aligned} \min \quad & \int_{x_a}^{x_b} \sqrt{\frac{1 + y'(x)^2}{y_a - y(x)}} dx \\ \text{sa} \quad & y(x_a) = y_a \\ & y(x_b) = y_b \end{aligned}$$

- Discreta Usual

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sqrt{\frac{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}{y_a - y_i}} \Delta x_{i-1} \\ \text{sa} \quad & y(x_a) = y_a \\ & y(x_b) = y_b \end{aligned}$$

A função L a considerar no algoritmo é

$$L(s, t, u, v, w) = \sqrt{\frac{1 + w^2}{y_a - v}}.$$

- Discreta de Guseinov

Para discretizar (à [4]), tendo em conta que a curva solução será aproximada por uma sequência de n segmentos de recta que ligam os pontos (x_{i-1}, y_{i-1}) e (x_i, y_i) , com $i = 1, \dots, n$ ($x_i > x_{i-1}$ e $y_{i-1} > y_i$), deve averiguar-se o tempo que uma partícula demora a percorrer cada um destes segmentos de recta (desprezando o atrito).

Sendo a_i a aceleração da partícula e θ_i o ângulo formado com o eixo dos x 's no segmento de recta i ,

$$a_i = g \sin \theta_i.$$

Usando a trigonometria tem-se que

$$\sin \theta_i = \frac{y_{i-1} - y_i}{\sqrt{(x_i - x_{i-1})^2 + (y_{i-1} - y_i)^2}} = -\frac{\Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (-\Delta y_{i-1})^2}},$$

em que $\Delta y_{i-1} = y_i - y_{i-1}$ e $\Delta x_{i-1} = x_i - x_{i-1}$.

$$\text{Assim, } a_i = -\frac{g \Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}}.$$

Como $a_i = \frac{dv_i}{dt}$ tem-se que $\frac{dv_i}{dt} = -\frac{g \Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}}$, pelo que

$$v_i = -\frac{g \Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}} t + c.$$

Além disso, se $t = 0$ então $c = v_0$. Mais ainda, em cada segmento de recta é reiniciada a contagem do tempo (contabiliza-se o tempo necessário para percorrer cada segmento de recta, somando-se todos os tempos no final). Desta forma, em cada segmento de recta, $t_{\text{inicial}} = 0$. Logo, $c = v_{i-1}$ (e $v_0 = 0$), donde

$$v_i = -\frac{g \Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}} t + v_{i-1}.$$

Por outro lado, $v = \frac{ds}{dt}$, donde

$$v dt = ds \Leftrightarrow \int_0^{t_i} v dt = \int ds,$$

sendo t_i o tempo que a partícula demora a percorrer o segmento de recta i .

$$\begin{aligned} & \int_0^{t_i} \left(-\frac{g\Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}} t + v_{i-1} \right) dt = \\ & -\frac{g\Delta y_{i-1}}{\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}} \int_0^{t_i} t dt + v_{i-1} \int_0^{t_i} dt = \\ & -\frac{g\Delta y_{i-1}}{2\sqrt{(\Delta x_{i-1})^2 + (\Delta y_{i-1})^2}} t_i^2 + v_{i-1} t_i = \\ & -\frac{g\Delta y_{i-1}}{2\Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}} t_i^2 + v_{i-1} t_i \end{aligned}$$

Para determinar o integral de linha ds usa-se a parametrização do segmento de recta que liga os pontos (x_{i-1}, y_{i-1}) e (x_i, y_i) :

$$\begin{aligned} x(t) &= t \\ y(t) &= \frac{\Delta y_{i-1}}{\Delta x_{i-1}} t + y_{i-1} - \frac{\Delta y_{i-1}}{\Delta x_{i-1}} x_{i-1}, t \in [x_{i-1}, x_i]. \end{aligned}$$

$$\begin{aligned} \int ds &= \int_{x_{i-1}}^{x_i} \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} dt \\ &= \int_{x_{i-1}}^{x_i} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} dt \\ &= \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \int_{x_{i-1}}^{x_i} dt \\ &= \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} (x_i - x_{i-1}) \\ &= \Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \end{aligned}$$

Tem-se então

$$\begin{aligned}
& -\frac{g\Delta y_{i-1}}{2\Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}t_i^2 + v_{i-1}t_i = \Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} \\
\Leftrightarrow & -\frac{g\Delta y_{i-1}}{2\Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}t_i^2 + v_{i-1}t_i - \Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} = 0 \\
\Leftrightarrow & \frac{g\Delta y_{i-1}}{2\Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}t_i^2 - v_{i-1}t_i + \Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2} = 0
\end{aligned}$$

Usando a Fórmula Resolvente

$$\begin{aligned}
t_i &= \frac{v_{i-1} \pm \sqrt{v_{i-1}^2 - 2g\Delta y_{i-1}}}{\frac{g\Delta y_{i-1}}{\Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}} \\
\Leftrightarrow t_i &= \frac{\Delta x_{i-1}\sqrt{1+\left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g\Delta y_{i-1}} \left(v_{i-1} \pm \sqrt{v_{i-1}^2 - 2g\Delta y_{i-1}} \right)
\end{aligned}$$

Como se pretende obter uma expressão que dependa apenas dos pontos usados para discretizar a curva, deve determinar-se a velocidade inicial da partícula em cada segmento de recta i (v_{i-1}) também em função destes valores. Usando a Lei de Conservação de Energia, o valor da Energia Mecânica no início do primeiro segmento de recta (E_{m_a}) é igual ao valor desta energia no início de cada um dos outros segmentos de recta. Considerando o segmento de recta i ,

$$E_{m_a} = E_{m_i}.$$

Num determinado momento, a Energia Mecânica da partícula é a soma da sua Energia Potencial (*energia em potência* tem a ver com a posição da partícula) e da sua Energia Cinética (relativa ao movimento da partícula). Sendo

g a aceleração gravítica,

m a massa da partícula,

h a altura a que está a partícula,

v a velocidade da partícula,

$$E_p = mgh \text{ e } E_c = \frac{mv^2}{2}.$$

Assim, no segmento de recta i (com extremos (x_{i-1}, y_{i-1}) e (x_i, y_i)),

$$\begin{aligned}
E_{m_a} &= E_{m_i} \\
\Leftrightarrow E_{p_a} + E_{c_a} &= E_{p_i} + E_{m_i} \\
\Leftrightarrow gmy_a + \frac{mv_0^2}{2} &= gmy_{i-1} + \frac{mv_{i-1}^2}{2} \\
\Leftrightarrow gy_a + \frac{v_0^2}{2} &= gy_{i-1} + \frac{v_{i-1}^2}{2}
\end{aligned}$$

Como a partícula parte do repouso, $v_0 = 0$,

$$\begin{aligned}
gy_a &= gy_{i-1} + \frac{v_{i-1}^2}{2} \\
\Leftrightarrow v_{i-1} &= \pm \sqrt{2g(y_a - y_{i-1})}
\end{aligned}$$

Além disso, $v_{i-1} \geq 0$ para qualquer $i = 1, 2, \dots, n$. Logo

$$v_{i-1} = \sqrt{2g(y_a - y_{i-1})}.$$

Substituindo na expressão encontrada para t_i

$$\begin{aligned}
t_i &= \frac{\Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g\Delta y_{i-1}} \left(\sqrt{2g(y_a - y_{i-1})} \pm \sqrt{2g(y_a - y_{i-1}) - 2g(y_i - y_{i-1})} \right) \\
\Leftrightarrow t_i &= \frac{\Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g\Delta y_{i-1}} \left(\sqrt{2g(y_a - y_{i-1})} \pm \sqrt{2g(y_a - y_i)} \right).
\end{aligned}$$

Como $\Delta y_{i-1} < 0$ e $\Delta x_{i-1} > 0$, então $\frac{\Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g\Delta y_{i-1}} < 0$. Por outro lado, como a contagem do tempo não pode ser negativa, a seguinte condição deve ser verdadeira

$$\sqrt{2g(y_a - y_{i-1})} \pm \sqrt{2g(y_a - y_i)} \leq 0.$$

$$\begin{aligned}
y_{i-1} > y_i &\Rightarrow y_a - y_{i-1} < y_a - y_i \\
&\Rightarrow \sqrt{2g(y_a - y_{i-1})} < \sqrt{2g(y_a - y_i)} \\
&\Rightarrow \sqrt{2g(y_a - y_{i-1})} - \sqrt{2g(y_a - y_i)} < 0.
\end{aligned}$$

Assim, a operação ‘+’ não pode ser considerada.

Logo, o tempo que uma partícula leva a percorrer o segmento de recta i é

$$\begin{aligned}
t_i &= \frac{\Delta x_{i-1} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g \Delta y_{i-1}} \left(\sqrt{2g(y_a - y_{i-1})} - \sqrt{2g(y_a - y_i)} \right) \\
&= \frac{\sqrt{2g} \sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{g \Delta y_{i-1}} (\sqrt{y_a - y_{i-1}} - \sqrt{y_a - y_i}) \Delta x_{i-1} \\
&= \sqrt{\frac{2}{g}} \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\Delta y_{i-1} (\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i})} (\sqrt{y_a - y_{i-1}} - \sqrt{y_a - y_i}) \\
&\quad \cdot (\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i}) \Delta x_{i-1} \\
&= \sqrt{\frac{2}{g}} \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\Delta y_{i-1} (\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i})} (y_a - y_{i-1} - y_a + y_i) \Delta x_{i-1} \\
&= \sqrt{\frac{2}{g}} \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\Delta y_{i-1} (\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i})} (-y_{i-1} + y_i) \Delta x_{i-1} \\
&= \sqrt{\frac{2}{g}} \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i}} \Delta x_{i-1}.
\end{aligned}$$

Somando os tempos dos n segmentos de recta tem-se

$$\sum_{i=1}^n t_i = \sqrt{\frac{2}{g}} \sum_{i=1}^n \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i}} \Delta x_{i-1}.$$

Para encontrar uma solução aproximada do Problema da Braquistócrona, procuram-se os y_i 's tais que

$$\min \sum_{i=1}^n \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i}} \Delta x_{i-1}.$$

O problema é então

$$\begin{aligned}
\min \quad & \sum_{i=1}^n \frac{\sqrt{1 + \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}}\right)^2}}{\sqrt{y_a - y_{i-1}} + \sqrt{y_a - y_i}} \Delta x_{i-1} \\
\text{sa} \quad & y(x_a) = y_a \\
& y(x_b) = y_b.
\end{aligned}$$

E a função a L a usar no algoritmo é

$$L(s, t, u, v, w) = \frac{\sqrt{1 + w^2}}{\sqrt{y_a - u} + \sqrt{y_a - v}}.$$

- Contínua para OPTRAGEN (formulação de Bolza)

$$\begin{aligned}
\min \quad & \int_{x_a}^{x_b} \sqrt{\frac{1+z(x)^2}{y_a-y(x)}} dx \\
sa \quad & y'(x) = z(x) \\
& y(x_a) = y_a \\
& y(x_b) = y_b
\end{aligned}$$

- Contínua para OC (formulação de Mayer)

$$\begin{aligned}
\min \quad & u(x_b) \\
sa \quad & y'(x) = z(x) \\
& u'(x) = \sqrt{\frac{1+z(x)^2}{y_a-y(x)}} \\
& u(x_a) = 0 \\
& y(x_a) = y_a \\
& y(x_b) = y_b.
\end{aligned}$$

3.3.2 Resultados

O algoritmo (§3.2 na página 38) apresenta aproximações de soluções de problemas do Cálculo das Variações através da discretização desses problemas. Assim, os resultados aqui apresentados dizem respeito aos dois tipos de discretização propostos: Discretização de Guseinov (página 7) e Discretização Usual (página 6).

Além disso, aquando da descrição do algoritmo e da ferramenta usado na sua implementação, referiu-se que é resolvido um problema de Programação Não Linear para obter a solução do sistema de Equações Não Lineares. Como um problema de Programação Não Linear pode ter várias restrições, foram testados dois conjuntos de restrições. Do primeiro conjunto (“Restrições 1”) fazem apenas parte as Condições de Fronteira do problema do Cálculo das Variações. As restrições do segundo conjunto (“Restrições 2”) já têm em conta a natureza do problema e, neste caso, além das Condições de Fronteira, considera-se que

$$|y_i| \leq \max\{|\alpha|, |\beta|\}, \quad i = 1, \dots, n-1. \quad (3.4)$$

São ainda aqui apresentados os resultados obtidos usando os quatro métodos para resolver o problema de Programação Não Linear: *Random Search* (RS), *Nelder Mead* (NM), *Differential Evolution* (DE) e *Simulated Annealing* (SA).

Considere-se o Problema da Braquistócrona (3.1) com as condições de fronteira

$$\begin{aligned}
x_a &= 0 \\
y_a &= 10 \\
x_b &= 10 \\
y_b &= 0
\end{aligned} \quad (3.5)$$

tem como solução óptima a função (*cf.* (3.2))

$$\begin{cases} x = 5.729170375\theta - 5.729170375 \sin(\theta) \\ y = 4.270829625 + 5.729170375 \cos(\theta) \end{cases}, \quad 0 \leq \theta \leq 2.412011144.$$

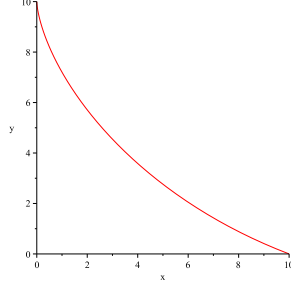


Figura 3.1: Gráfico da curva de tempo mínimo entre os pontos (0, 10) e (10, 0).

O valor óptimo, não considerando a constante $\frac{1}{\sqrt{2g}}$, é, atendendo a (3.3),

$$\int_0^{2.412011144} \sqrt{\frac{1 + \left(\frac{32.82339319 \sin(\theta)}{5.729170375 - 5.729170375 \cos(\theta)} \right)^2}{5.729170375 - 5.729170375 \cos(\theta)}} (5.729170375 - 5.729170375 \cos(\theta)) d\theta = 8.164698962.$$

A Tabela 3.1 apresenta os valores do integral (3.3) usando como funções integrandas as funções definidas por ramos lineares a partir das aproximações à solução de (3.1) encontradas pelo algoritmo proposto em §3.2 e a função definida por ramos lineares a partir da solução óptima (FROpt), cujo valor pode ser usado como referência.

Tabela 3.1: Valores de $\int_{x_a}^{x_b} f(x, \bar{y}(x), \bar{y}'(x))dx$ para o problema 3.1 em que $x_a = 0$, $x_b = 10$, $y_a = 10$ e $y_b = 0$.

Nº Int	Mét	Disc. Guseinov		Disc. Usual		Integral
		Restrições 1 (3.5)	Restrições 2 (3.4 e 3.5)	Restrições 1	Restrições 2	FROpt
3	RS	8.353400	8.353400	8.417810	8.417810	8.36926
	NM	8.353400	8.353400	8.417810	8.417810	
	DE	*	*	*	8.417810	
	SA	8.353400	8.353400	8.417810	8.417810	
5	RS	8.271470	8.271470	8.464080	8.464080	8.28066
	NM	8.271470	8.271470	8.464080	8.464080	
	DE	*	*	*	*	
	SA	8.271470	**	8.464080	8.464080	
8	RS	8.229050	8.229050	8.565700	8.565700	8.23454
	NM	8.229050	9.037230	8.565700	9.906430	
	DE	*	*	*	*	
	SA	***	8.229050	8.565700	12.287800	
10	RS	$7.4995 \times 10^8(***)$	12.716400	8.617300	12.681700	8.21981
	NM	8.215500	12.716400	8.617300	8.617250	
	DE	*	*	*	*	
	SA	***	12.716400	8.617300	15.370300	
15	RS	***	**	8.701864	12.749044	8.20069
	NM	***	12.763500	8.701864	8.701870	
	DE	*	*	*	*	
	SA	*	*	8.701860	*	
20	RS	8.189344	**	8.751132	12.782401	8.19139
	NM	***	15.972220	8.751132	21.052693	
	DE	*	25.114541	*	*	
	SA	***	*	*	*	

‘*’: o método usado para resolver o problema de Programação Não Linear não termina a sua execução com sucesso (por exemplo, excedeu o limite de iterações);

‘**’: valor encontrado é um número complexo com parte imaginária desprezável;

‘***’: valor encontrado muito elevado e cuja solução não faz sentido (Observação 3.3.1).

Observação 3.3.1. *Alguns métodos não indicam qualquer irregularidade na sua execução mas apresentam valores do funcional que não fazem sentido, tendo em conta o valor óptimo. Por exemplo, para $n = 10$ na discretização de Guseinov e considerando apenas as Condições de Fronteira como restrições do problema de Programação Não Linear, o método Random Search tem valor do funcional $7,4995 \times 10^8$ a partir da solução*

$$\begin{aligned} y_0 &= 10; \\ y_1 &= y_2 = y_3 = -1,02157 \times 10^{16}; \\ y_4 &= y_5 = y_6 = -1,21949 \times 10^{16}; \\ y_7 &= y_8 = y_9 = -3,51518 \times 10^{16}; \\ y_{10} &= 0. \end{aligned}$$

Esta solução não é aceitável como aproximação para o problema proposto.

Para impedir que surgissem soluções como esta acrescentaram-se às restrições as condições $|y_i| \leq \max\{|\alpha|, |\beta|\}, i = 1, \dots, n - 1$ (“Restrições 2”). No entanto, de um modo geral, este condicionamento adicional não surtiu o efeito pretendido (ver Tabela 3.1).

Apresentam-se, nas Figuras 3.2 a 3.12, as curvas definidas a partir das soluções aproximadas encontradas pelo algoritmo. Observando a Tabela 3.1 conclui-se que, de um modo geral, os valores dos integrais dos problemas em que se consideram apenas as condições de fronteira (“Restrições 1”) são melhores do que os valores dos mesmos integrais considerando também outras restrições, apenas se apresenta, para cada n a melhor solução para este conjunto de restrições.

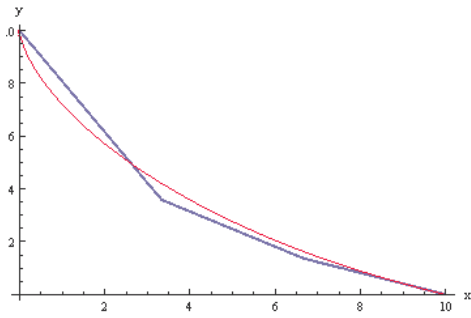


Figura 3.2: Curva encontrada aplicando a discretização de Guseinov, com $n = 3$ e o método *Nelder Mead*.

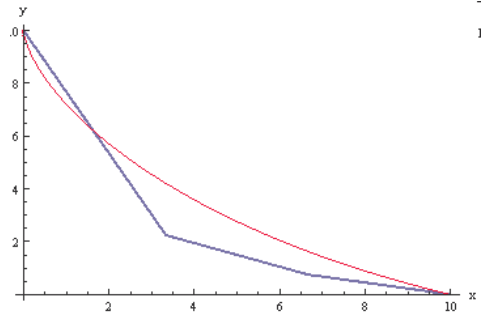


Figura 3.3: Curva encontrada aplicando a discretização usual, com $n = 3$ e o método *Random Search*.

Como se observa, com a discretização proposta por Guseinov obtêm-se melhores soluções do que usando a discretização usual. Isto deve-se ao facto da primeira discretização ter em conta a natureza do problema e o modo como a aproximação à solução óptima é efectuada.

Além disso, usando a discretização de Guseinov, as soluções melhoram com o aumento do número de intervalos em que se divide o domínio de integração.

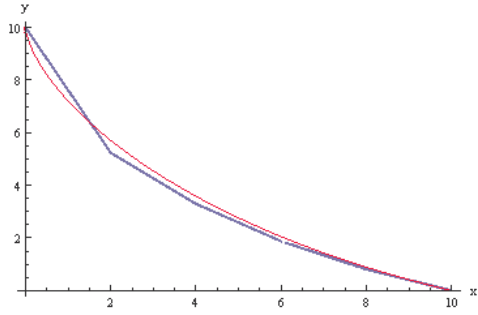


Figura 3.4: Curva encontrada aplicando a discretização de Guseinov, com $n = 5$ e o método *Random Search*.

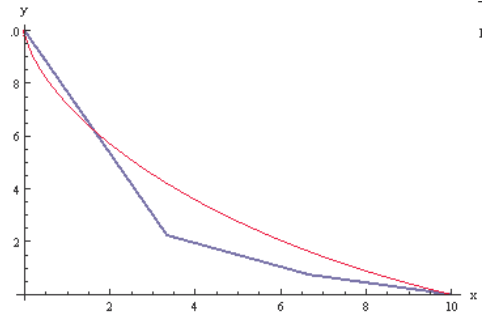


Figura 3.5: Curva encontrada aplicando a discretização usual, com $n = 5$ e o método *Random Search*.

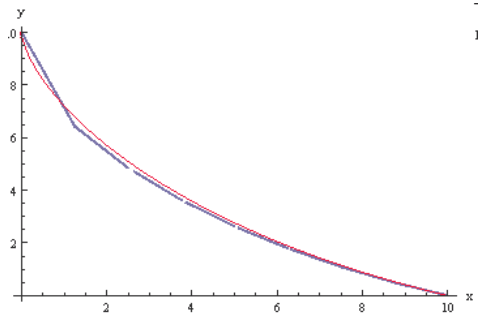


Figura 3.6: Curva encontrada aplicando a discretização de Guseinov, com $n = 8$ e o método *Random Search*.

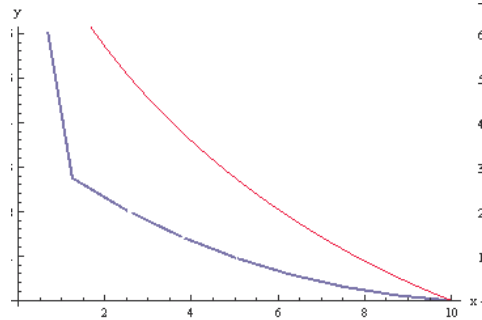


Figura 3.7: Curva encontrada aplicando a discretização usual, com $n = 8$ e o método *Random Search*.

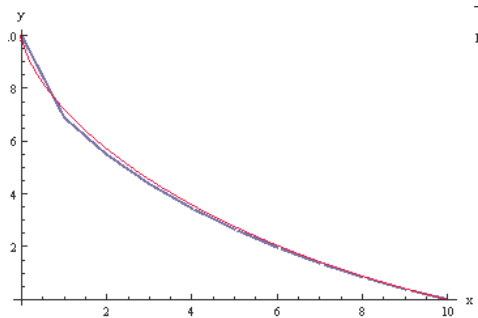


Figura 3.8: Curva encontrada aplicando a discretização de Guseinov, com $n = 10$ e o método *Nelder Mead*.

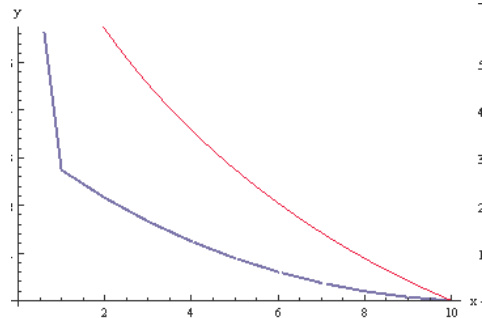


Figura 3.9: Curva encontrada aplicando a discretização usual, com $n = 10$ e o método *Random Search*.

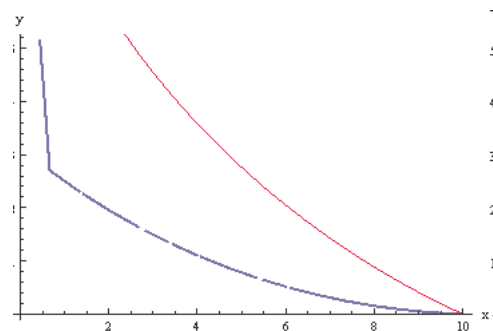


Figura 3.10: Curva encontrada aplicando a discretização usual, com $n = 15$ e o método *Random Search*.

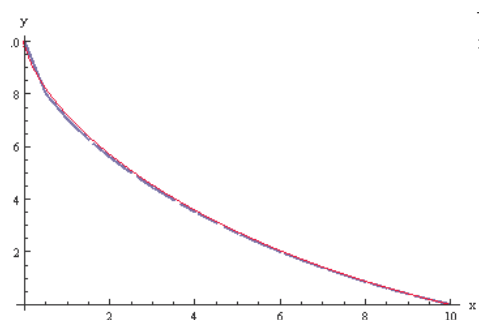


Figura 3.11: Curva encontrada aplicando a discretização de Guseinov, com $n = 20$ e o método *Random Search*.

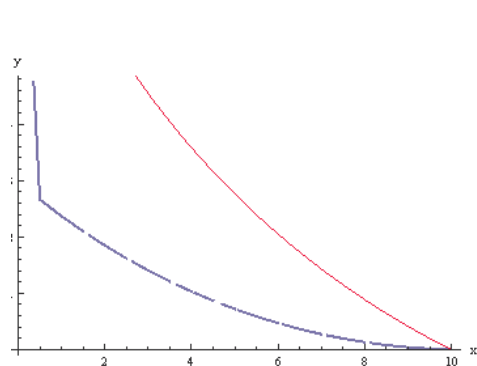


Figura 3.12: Curva encontrada aplicando a discretização usual, com $n = 20$ e o método *Random Search*.

3.3.3 Estudo Comparativo

Na secção anterior apresentaram-se os resultados obtidos para o Problema da Braquistócrona com a implementação do algoritmo proposto por Guseinov([4]) no *Mathematica*. No entanto, muitas outras estratégias já foram aplicadas para resolver este problema: métodos tradicionais que decorrem do Cálculo Diferencial, técnicas de optimização numérica, entre outros.

Estas técnicas procuram, por vezes, dar resposta a pequenas variantes do problema que dizem respeito, por exemplo, a pontos pelos quais deve passar a curva solução do problema ou à consideração do atrito no problema.

Em [15], a autora procura a solução para o Problema da Braquistócrona usando o *Maple* para determinar e resolver a Equação de Euler-Lagrange. No entanto, como esta equação é muito complexa, considera a representação paramétrica do problema e trabalha com o integral de energia para encontrar a solução óptima. Assim, neste texto são apresentadas duas resoluções analíticas para o problema, apesar de apenas uma fornecer a solução, e alguns exemplos.

Também os autores de [10] apresentam uma resolução do Problema da Braquistócrona, desta feita, a mesma sugerida por Johann Bernoulli (aquando da sua proposta do problema). Esta estratégia aplica o Princípio de Fermat, usado em problemas de Óptica (“a luz propaga-se de um ponto para outro em tempo mínimo”), a um problema de Mecânica como é o Problema da Braquistócrona.

Os autores de [3] aplicam, como já se referiu (página 33), um algoritmo de estratégias evolucionistas combinadas com Controlo Óptimo em vários exemplos do Problema da Braquistócrona. As soluções encontradas por este algoritmo são também funções definidas por ramos. Para o problema aqui testado (curva de tempo mínimo entre os pontos (0,10) e (10,0)), a implementação deste algoritmo determina uma curva com 20 segmentos de recta, de tempo 8,19365 (ignorando a constante multiplicativa do integral).

A ferramenta OC (página 32) pode ser usada em problemas do Cálculo das Variações formulados como problemas de Controlo Óptimo na forma de Mayer. Para obter soluções para estes problemas pode usar-se qualquer dos exercícios/exemplos (em C++) fornecidos pelos autores e alterar os dados relativos ao enunciado do problema, guardar com um novo nome (por exemplo, *problema1.cpp*), compilar o novo programa (os autores propõem o *Borland C++* – linha de comandos: *bcc32 problema1.cpp*) e criar o executável para a aplicação, um ficheiro do tipo *ocp* (linha de comandos: *Comp problema1*). Para obter a solução, “carrega-se” o ficheiro *ocp* através da instrução *Load Problem* no menu *File* da ferramenta OC.

As várias opções (por exemplo, o método para resolver problema de Programação Matemática, o método para resolver Equações Diferenciais, o número de iterações, a solução inicial) podem ser escolhidas e alteradas no interface da aplicação, sem recorrer à programação em C++.

É possível continuar a aplicar os métodos excedendo o número de iterações inicialmente

definido. Quando é atingida a precisão, também definida, a execução termina. No entanto, por vezes ainda é possível continuar a execução mas a diferença entre cada elemento das soluções consecutivas é muito pequena obrigando a que sejam efectuadas muitas iterações até que se encontre uma solução consideravelmente melhor.

Cada execução (considerada a partir do momento em que se carrega no botão “Start”) pode terminar com diferentes comentários: *Iterations limit* (limite de iterações atingido), *Accuracy reached* (precisão atingida), *Use another method* (usar outro método) ou *Invalid floating point operation* (operação envolve um cálculo que não pode ser efectuado). Apenas na última situação não é apresentada qualquer solução para o problema.

As soluções encontradas pela ferramenta OC para o Problema da Braquistócrona usando diferentes algoritmos são apresentadas no Anexo D.1. É possível, como já se referiu, escolher em quantos intervalos se divide o domínio de integração do problema. No entanto, como as soluções melhoram com o aumento do número de intervalos, optou-se apresentar os resultados obtidos com $n = 20$.

Apresenta-se, na Tabela 3.2, o tempo que uma partícula demora a percorrer cada uma das curvas definidas a partir das soluções encontradas (Anexo D.1).

A melhor solução encontrada por esta ferramenta discretizando o domínio de integração do problema em 20 intervalos usou os métodos Gradientes Conjugados (CG) e *Runge-Kutta* 2 (R-K 2) (ver Figura 3.13).

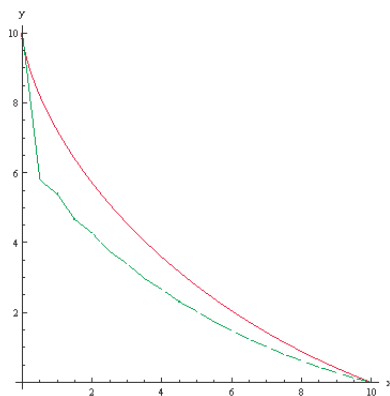


Figura 3.13: Gráfico da melhor solução encontrada pela ferramenta OC (verde) e gráfico da solução ótima (vermelho).

Tabela 3.2: Valores dos integrais das soluções encontradas pela ferramenta OC para $n = 20$.

Mét Opt	Mét ED		N.º it	Comentário	Integral
US	Euler	PL	301	precisão atingida	18.458960731546
		PC	401	precisão atingida	18.488987534126
	R-K 2	PL	4401	precisão atingida	8.3442763799772
		PC	1422	precisão atingida	8.4522489207825
DS 1	Euler	PL	600	precisão atingida	18.1327243940062
		PC	600	precisão atingida	18.1327243940062
	R-K 2	PL	2100	precisão atingida	8.3372098456220
		PC	446	precisão atingida	8.4433291523813
DS 2	Euler	PL	570	precisão atingida	18.1329568408810
		PC	570	precisão atingida	18.1329568408810
	R-K 2	PL	2002	precisão atingida	8.3371053840606
		PC	406	precisão atingida	8.4420997693540
RS	Euler	PL	5000	limite de iterações atingido	18.1312361102421
		PC	5000	limite de iterações atingido	18.1095791729667
	R-K 2	PL	5000	limite de iterações atingido	8.3470035977603
		PC	4018	precisão atingida	8.4430267037026
CG	Euler	PL	255	precisão atingida	18.1321186580871
		PC	255	precisão atingida	18.1321186580871
	R-K 2	PL	137	precisão atingida	8.3360784081642
		PC	47	precisão atingida	8.4429467474810
NM	Euler	PL	356	usar outro método	18.5106796365223
		PC	761	precisão atingida	18.1321051433615
	R-K 2	PL	81	usar outro método*	17.1211556690379
		PC	14	usar outro método*	39.4273

“*”: fazendo, por exemplo, de 10 em 10 iterações, estes métodos convergem

3.4 Exemplo 2: Problema com Fenómeno de Lavrentiev

3.4.1 Apresentação do Problema

O Fenómeno de Lavrentiev foi descoberto por este matemático em 1926. Os problemas do Cálculo das Variações que verificam este fenómeno apresentam diferentes soluções óptimas no conjunto de funções absolutamente contínuas³ e no conjunto de funções Lipschitzianas⁴, mais especificamente, o ínfimo do funcional no conjunto das funções absolutamente contínuas (num intervalo) é menor do que o ínfimo do mesmo funcional no conjunto das funções Lipschitzianas, o que dificulta a determinação computacional do mínimo do funcional.

O problema mais conhecido foi proposto por Manià (em 1934) e procura a função y que minimiza

$$I[y] = \int_0^1 (y(x)^3 - x)^2 (y'(x))^6 dx \quad (3.6)$$

com as condições de fronteira

$$y(0) = 0 \text{ e } y(1) = 1.$$

A Solução Óptima deste problema é a função

$$\hat{y}(x) = x^{\frac{1}{3}}$$

que está representada graficamente na Figura 3.14.

Ora, para qualquer $x \in [0, 1]$

$$L(x, y(x), y'(x)) = (y(x)^3 - x)^2 (y'(x))^6 \geq 0$$

e também

$$I[y] \geq 0.$$

Como

$$\forall x \in [0, 1] \quad L(x, \hat{y}(x), \hat{y}'(x)) = (\hat{y}(x)^3 - x)^2 (\hat{y}'(x))^6 = 0,$$

$$\int_0^1 (x - x)^2 \left(\frac{x^{-\frac{2}{3}}}{3} \right)^6 dx = 0,$$

³Uma função $f : [a, b] \rightarrow \mathbb{R}$ é *absolutamente contínua* em $[a, b]$, denotando-se por $f \in AC([a, b], \mathbb{R})$, se para qualquer $\epsilon > 0$ existe um $\delta > 0$ tal que para qualquer família finita de intervalos disjuntos dois a dois $]a_i, b_i[\subset [a, b]$, $i = 1, 2, \dots, n$,

$$\sum_{i=1}^n (b_i - a_i) < \delta \Rightarrow \sum_{i=1}^n |f(b_i) - f(a_i)| < \epsilon.$$

⁴Uma função $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ é *Lipschitziana* se existe uma constante $c > 0$ tal que $\forall x_i, x_j \in I$

$$|f(x_i) - f(x_j)| \leq c|x_i - x_j|.$$

Se $I =]a, b[$, com $a, b \in \mathbb{R}$, $f \in Lip(I, \mathbb{R})$.

f é *localmente Lipschitziana* se para qualquer $x \in I$ existe uma vizinhança de x na qual f é Lipschitziana.

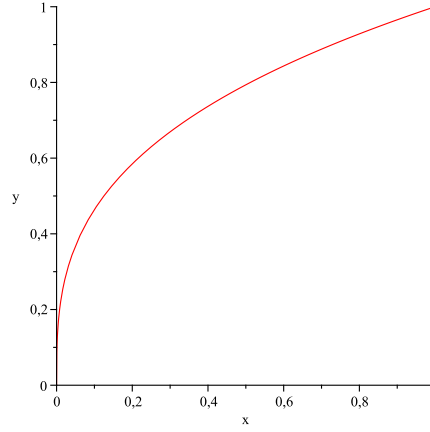


Figura 3.14: Gráfico da Solução Óptima do exemplo proposto por Manià.

ou seja, o valor óptimo deste problema é 0.

Verifique-se então, para o exemplo de Manià, o Fenómeno de Lavrentiev ([14],[17]), isto é, observe-se que

$$\inf_{y \in W_1} I(y) > \inf_{y \in W_2} I(y)$$

onde

$$W_1 = \{y \in Lip([0, 1]) : y(0) = 0, y(1) = 1\}$$

$$W_2 = \{y \in AC([0, 1]) : y(0) = 0, y(1) = 1\}.$$

A função $y(x) = x^{\frac{1}{3}}$ é absolutamente contínua em $[0, 1]$ e verifica as condições de fronteira ($y(x) \in W_1$). Considere-se um conjunto de n intervalos abertos $]a_i, b_i[$ disjuntos tais que $]a_i, b_i[\subset [0, 1], i = 1, 2, \dots, n$ e $\sum_{i=1}^n (b_i - a_i) < \delta$ (com $\delta > 0$).

Pelo Teorema do Valor Médio⁵ existe um $c_i \in]a_i, b_i[$ tal que $b_i^{\frac{1}{3}} - a_i^{\frac{1}{3}} = \frac{c_i^{-\frac{2}{3}}}{3} (b_i - a_i)$.

$$\sum_{i=1}^n |y(b_i) - y(a_i)| = \sum_{i=1}^n \left| b_i^{\frac{1}{3}} - a_i^{\frac{1}{3}} \right| = \sum_{i=1}^n \left| \frac{c_i^{-\frac{2}{3}}}{3} (b_i - a_i) \right| = \sum_{i=1}^n \frac{c_i^{-\frac{2}{3}}}{3} (b_i - a_i)$$

pois $\forall x \in]0, 1[\quad y'(x) > 0$.

Considerando $M = \max_{x \in]0, 1[} \{y'(x)\}$, tem-se que

$$\sum_{i=1}^n |y(b_i) - y(a_i)| < M\delta.$$

Assim, para qualquer ϵ existe um $\delta = \frac{\epsilon}{M}$ tal que $\sum_{i=1}^n (b_i - a_i) < \delta \Rightarrow \sum_{i=1}^n |y(b_i) - y(a_i)| < \epsilon$.

⁵ Teorema do Valor Médio ou de Lagrange:

Seja f uma função contínua em $[a, b]$ e diferenciável em $]a, b[$, $\exists c \in]a, b[$ tal que $f'(c) = \frac{f(b) - f(a)}{b - a}$.

Tenha-se agora em conta o conjunto das funções Lipschitzianas

$$W = \left\{ y \in Lip([0, 1]) : 0 \leq y(x) \leq \frac{x^{\frac{1}{3}}}{2} \right\}.$$

Para qualquer $x \in [0, 1]$ (considerando a segunda desigualdade)

$$L(x, y(x), y'(x)) = (y(x)^3 - x)^2 y'(x)^6 \geq \left(\left(\frac{x^{\frac{1}{3}}}{2} \right)^3 - x \right)^2 y'(x)^6 = \frac{7^2}{2^6} x^2 y'(x)^6.$$

Assim

$$I[y] = \int_0^1 (y(x)^3 - x)^2 y'(x)^6 dx \geq \int_0^1 \frac{7^2}{2^6} x^2 y'(x)^6 dx = \frac{7^2}{2^6} \int_0^1 x^2 y'(x)^6 dx.$$

Efectuando a mudança de variável $t = x^{\frac{3}{5}}$ e $z(t) = y(x)$,

$$\begin{aligned} \int_0^1 x^2 y'(x)^6 dx &= \int_0^{1^{\frac{3}{5}}} \left(t^{\frac{5}{3}} \right)^2 \left(\frac{3}{5} \left(t^{\frac{5}{3}} \right)^{-\frac{2}{5}} z'(t) \right)^6 \frac{5}{3} \left(t^{\frac{5}{3}} \right)^{-\frac{2}{5}} dt \\ &= \frac{3^5}{5^5} \int_0^1 t^{\frac{10}{3}} t^{-\frac{12}{3}} z'(t)^6 t^{\frac{2}{3}} dt \\ &= \frac{3^5}{5^5} \int_0^1 z'(t)^6 dt. \end{aligned}$$

Substituindo no cálculo anterior, tem-se

$$I(y) \geq \frac{3^5 7^2}{2^6 5^5} \int_0^1 z'(t)^6 dt.$$

Aplicando, neste integral, a Desigualdade de Hölder⁶ com $f(t) = z'(t)$, $g(t) = 1$, $p = 6$, $q = \frac{6}{5}$, $a = 0$ e $b = 1$, obtém-se

$$\int_0^1 |z'(t)| dt \leq \left[\int_0^1 |z'(t)|^6 dt \right]^{\frac{1}{6}} \left[\int_0^1 |1|^{\frac{6}{5}} dt \right]^{\frac{5}{6}}.$$

⁶Desigualdade de Hölder:

Sejam f e g duas funções reais de variável real definidas e integráveis sobre $[a, b]$, $1 < p, q < +\infty$ que verificam

$$\frac{1}{p} + \frac{1}{q} = 1$$

então

$$\int_a^b |f(x)g(x)| dx \leq \left[\int_a^b |f(x)|^p dx \right]^{\frac{1}{p}} \left[\int_a^b |g(x)|^q dx \right]^{\frac{1}{q}}.$$

Como $z'(t) \geq 0$ para qualquer $t \in [0, 1]$ (porque $y(x), y'(x) \geq 0$ para qualquer $x \in [0, 1]$),

$$\int_0^1 z'(t) dt \leq \left[\int_0^1 z'(t)^6 dt \right]^{\frac{1}{6}} \left[\int_0^1 1 dt \right]^{\frac{5}{6}}$$

$$\Leftrightarrow (z(1) - z(0))^6 \leq \int_0^1 z'(t)^6 dt (1 - 0)^{\frac{5}{6}}$$

$$\Leftrightarrow \int_0^1 z'(t)^6 dt \geq (z(1) - z(0))^6 = \left(y(1^{\frac{3}{5}}) - y(0^{\frac{3}{5}}) \right)^6 = (y(1) - y(0))^6 = (1 - 0)^6 = 1.$$

Logo

$$I[y] \geq \frac{3^5 7^2}{2^6 5^5} > 0.$$

Assim se verificou que o valor do integral no conjunto das funções Lipschitzianas é maior do que o valor do integral no conjunto das funções absolutamente contínuas (no intervalo considerado), para o exemplo de Manià.

Apesar da solução ótima deste problema ser simples, pode não ser fácil encontrá-la.

Formulações do Problema que verifica o Fenómeno de Lavrentiev

- Contínua

$$\begin{aligned} \min \quad & \int_0^1 (y(x)^3 - x)^2 (y'(x))^6 dx \\ \text{sa} \quad & y(0) = 0 \\ & y(1) = 1 \end{aligned}$$

- Discreta Usual

$$\begin{aligned} \min \quad & \sum_{i=1}^n (y_i^3 - x_i)^2 \left(\frac{\Delta y_{i-1}}{\Delta x_{i-1}} \right)^6 \Delta x_{i-1} \\ \text{sa} \quad & y(0) = 0 \\ & y(1) = 1 \end{aligned}$$

A função f a considerar no algoritmo é

$$f(s, t, u, v, w) = (v^3 - t)^2 w^6.$$

- Discreta de Guseinov

Como este problema é um exemplo teórico e não se conhece o seu significado físico, não é possível construir a discretização proposta por Guseinov.

- Contínua para OPTRAGEN (Formulação de Bolza)

$$\begin{aligned} \min \quad & \int_0^1 (y(x)^3 - x)^2 (z(x))^6 dx \\ \text{sa} \quad & y'(x) = z(x) \\ & y(0) = 0 \\ & y(1) = 1 \end{aligned}$$

- Contínua para OC (Formulação de Mayer)

$$\begin{aligned}
\min \quad & u(1) \\
\text{sa} \quad & y'(x) = z(x) \\
& u'(x) = (y(x)^3 - x)^2 (z(x))^6 \\
& u(0) = 0 \\
& y(0) = 0 \\
& y(1) = 1.
\end{aligned}$$

3.4.2 Resultados

Aqui apresentam-se os resultados obtidos pelo algoritmo introduzido na secção 3.2 usando a discretização usual com dois conjuntos de restrições no Problema de Programação Não Linear. O primeiro conjunto de restrições contém apenas as Condições de Fronteira do problema. O segundo conjunto tem, além das Condições de Fronteira, as restrições

$$0 \leq y_i \leq \max\{|\alpha|, |\beta|\}, i = 0, \dots, n.$$

Os resultados são referentes aos diferentes métodos usados para encontrar soluções do Problema de Programação Não Linear.

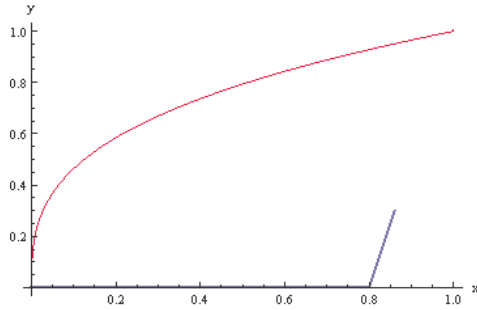


Figura 3.15: Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 5$ e o método *Nelder Mead*.

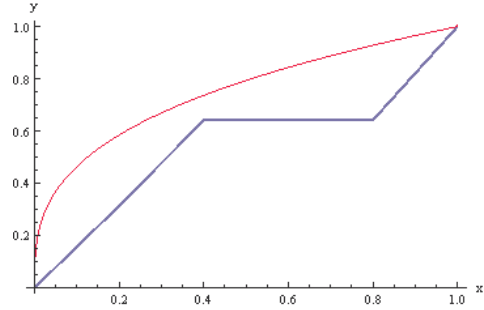


Figura 3.16: Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 5$ e o método *Nelder Mead*.

Como se observa na Tabela 3.3, acrescentando ao problema o segundo conjunto de restrições, obtêm-se, de um modo geral, melhores soluções, no sentido de que o valor do integral da curva definida por esses pontos diminui usando o mesmo método (ver, por exemplo, as Figuras 3.15 e 3.16).

Note-se ainda que, segundo a mesma tabela, a melhor solução foi encontrada usando o Método *Simulated Annealing*, com 8 intervalos e considerando todas as restrições, sendo o valor do integral 0.327614 (Figura 3.18). No entanto, visualizando os gráficos da Figura

Tabela 3.3: Valores de $\int_{x_a}^{x_b} L(x, \bar{y}(x), \bar{y}'(x))dx$ para o problema em que $x_a = 0$, $x_b = 1$, $y_a = 0$ e $y_b = 1$.

Nº de Intervalos	Método	Discretização Usual		Integral FROpt
		Restrições 1	Restrições 2	
3	RS	92.314300	92.314300	0.228573
	NM	92.314300	92.314300	
	DE	92.314300	92.314300	
	SA	92.314300	92.314300	
5	RS	1488.100000	1488.100000	0.380955
	NM	1488.090000	0.993566	
	DE	1488.100000	1488.100000	
	SA	1488.100000	1488.100000	
8	RS	17549.400000	14.767600	0.609528
	NM	539.515000	537.198000	
	DE	64.207400	17549.400000	
	SA	17549.400000	0.327614	
10	RS	55619.100000	0.490090	0.76191
	NM	55619.100000	20.364400	
	DE	2390.020000	55619.000000	
	SA	55619.100000	3.749560	
15	RS	443732.000000	1.156490	1.14287
	NM	443732.000000	12416.200000	
	DE	0.473695	443732.000000	
	SA	443732.000000	66.265100	
20	RS	1915810.000000	127.346000	1.52382
	NM	1915810.000000	10.174300	
	DE	2235.890000	32784.700000	
	SA	0.956401	7.473450	

3.20 à Figura 3.24, as curvas definidas pelas soluções com mais intervalos de discretização ($n = 10, n = 15$ e $n = 20$) parecem aproximar-se mais da curva óptima, como seria esperado.

Ainda na Tabela 3.3, observa-se que o valor dos funcionais calculados com as funções definidas por ramos lineares a partir da solução óptima aumenta com o número de intervalos. Este facto, apesar de parecer contraditório, não o é, trata-se de mais uma consequência do Fenómeno de Lavrentiev verificado por este exemplo. Veja-se o seguinte teorema proposto em [14].

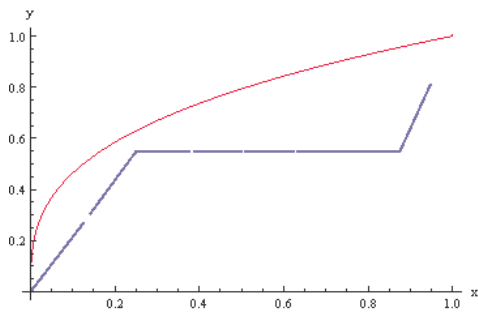


Figura 3.17: Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 8$ e o método *Differential Evolution*.

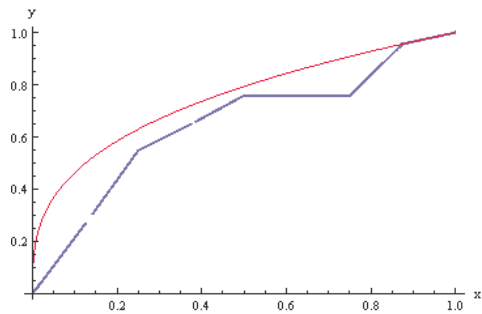


Figura 3.18: Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 8$ e o método *Simulated Annealing*.

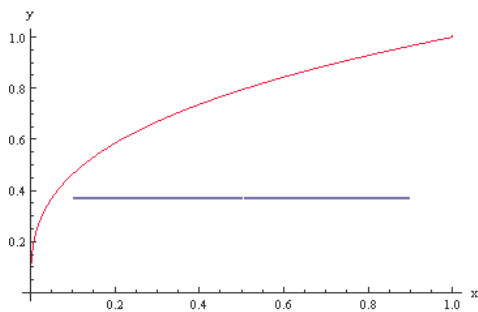


Figura 3.19: Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 10$ e o método *Differential Evolution*.

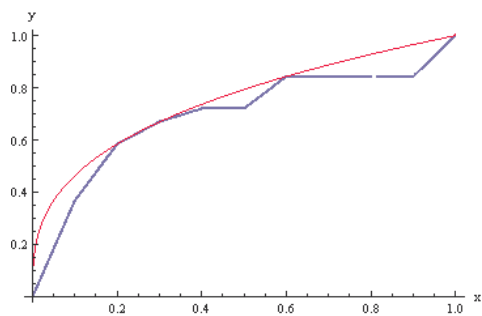


Figura 3.20: Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 10$ e o método *Random Search*.

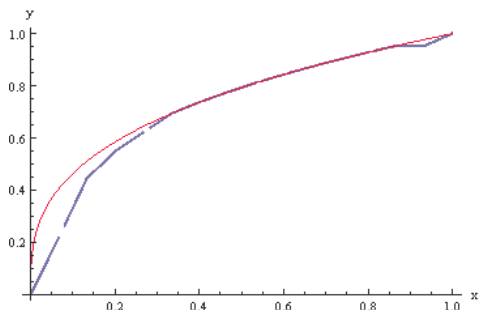


Figura 3.21: Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 15$ e o método *Differential Evolution*.

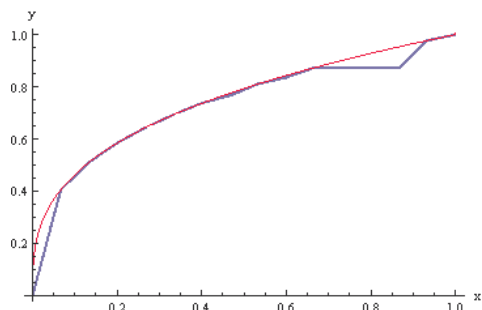


Figura 3.22: Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 15$ e o método *Random Search*.

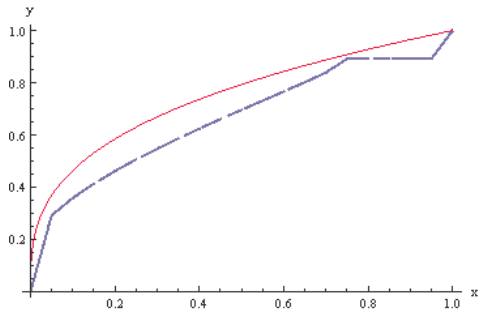


Figura 3.23: Curva encontrada aplicando a discretização usual com “Restrições 1”, $n = 20$ e o método *Simulated Annealing*.

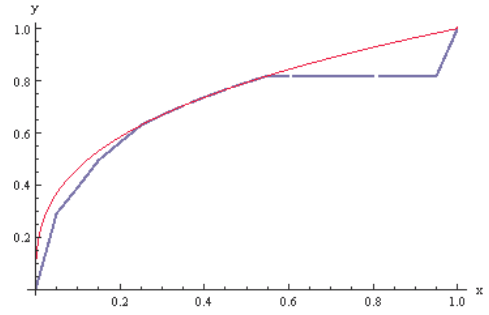


Figura 3.24: Curva encontrada aplicando a discretização usual com “Restrições 2”, $n = 20$ e o método *Simulated Annealing*.

Teorema 3.4.1. *Para qualquer sucessão de trajectórias de Lipschitz $\{y_n\}_n$ tal que y_n tende para \hat{y} quando n tende para ∞ , para quase todo $x \in [0, 1]$ tem-se que $I(y_n)$ (definido em 3.6) tende para ∞ .*

Demonstração. Ver [14].

□

3.4.3 Estudo Comparativo

Ao contrário do Problema da Braquistócrona, o exemplo teórico proposto por Manià não é muito estudado. No entanto, alguns métodos já foram elaborados para dar resposta a problemas que verificam o Fenómeno de Lavrentiev.

Em [1] é proposto um método de truncatura que, de forma muito simplista, consta na determinação de um limite superior para o mínimo do integral (já apresentado) através da utilização de um funcional auxiliar cujo domínio de integração está contido no domínio de integração original de forma a “contornar” os inconvenientes do Fenómeno de Lavrentiev. Também estes novos domínios de integração são divididos em n intervalos.

Os resultados numéricos apresentados no artigo ([1]) não incluem o valor do integral usado aqui como termo de comparação. No entanto, a observação dos gráficos obtidos sugere que as soluções são muito razoáveis, mantendo-se a ressalva de como, neste problema, uma razoável aproximação gráfica, nem sempre se reflecte no valor do integral.

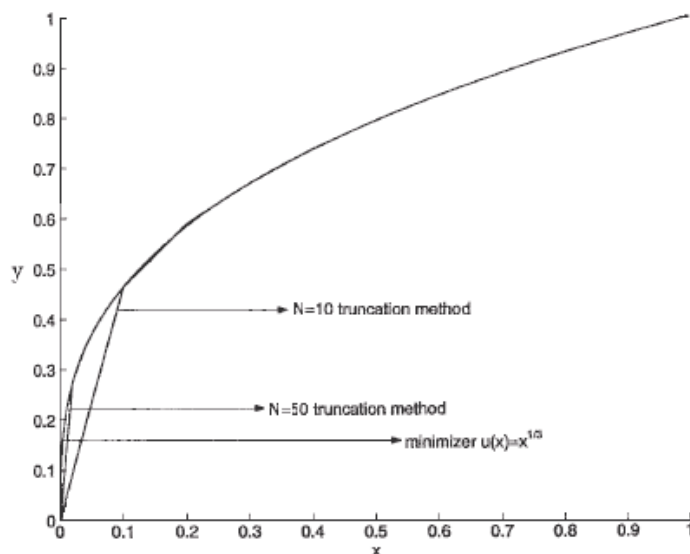


Figura 3.25: Gráfico com soluções encontradas pelo método de truncatura.

A ferramenta OC (página 32) apresenta soluções usando os diferentes métodos excepto com o Método de Newton (*Newton Method*). De seguida apresenta-se a tabela com os resultados obtidos com cada combinação de métodos considerando 20 intervalos.

A Figura 3.41 apresenta a curva cujo integral tem valor mais próximo do 0 (valor óptimo). No entanto, e como já se viu, esta curva pode não ser a mais próxima da curva óptima. Vejam-se então os gráficos da Figura 3.26 à Figura 3.41 com as diferentes soluções.

Tabela 3.4: Valores dos integrais das soluções encontradas pela ferramenta OC (página 32) para $n = 20$.

Mét Opt	Mét ED		N.º it	Comentário	Integral
US	Euler	PL	1	precisão atingida	395810
		PC	1	precisão atingida	395810
	R-K 2	PL	1501	precisão atingida	1.45193
		PC	4001	precisão atingida	1361.43
DS 1	Euler	PL	502	precisão atingida	5.8115
		PC	502	precisão atingida	5.8115
	R-K 2	PL	4200	precisão atingida	45.5153
		PC	78	precisão atingida	0.032702
DS 2	Euler	PL	300	precisão atingida	1.96464
		PC	300	precisão atingida	1.96464
	R-K 2	PL	3401	precisão atingida	47.0389
		PC	54	precisão atingida	0.032702
RS	Euler	PL	5000	limite de iterações atingido	1.52609
		PC	3921	precisão atingida	1.52387
	R-K 2	PL	5000	limite de iterações atingido	0.0329902
		PC	750	precisão atingida	0.0327021
CG	Euler	PL	13	precisão atingida	759.959
		PC	13	precisão atingida	759.959
	R-K 2	PL	343	precisão atingida	59.5272
		PC	23	precisão atingida	0.0326998
NM	Euler	PL	—	operação inválida	—
		PC	—	operação inválida	—
	R-K 2	PL	—	operação inválida	—
		PC	—	operação inválida	—

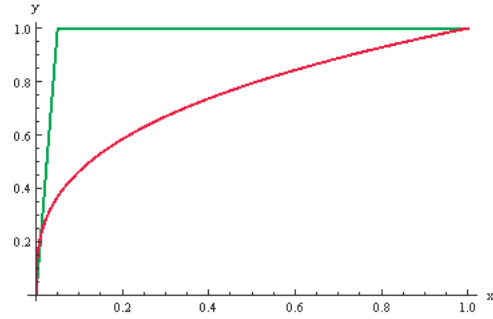


Figura 3.26: Solução obtida pelos métodos “Pesquisa Unidireccional”, Euler, com *Piecewise Linear* que coincide com a solução obtida pelos métodos “Pesquisa Unidireccional”, Euler, com *Piecewise Constant*.

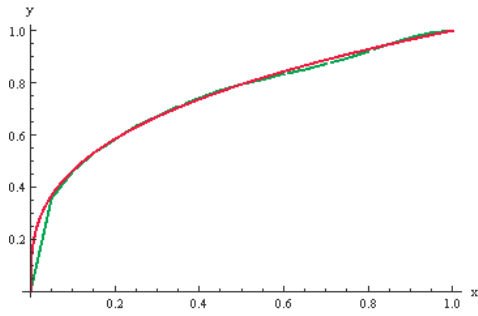


Figura 3.27: Solução obtida pelos métodos “Pesquisa Unidireccional”, Runge-Kutta, com *Piecewise Linear*.

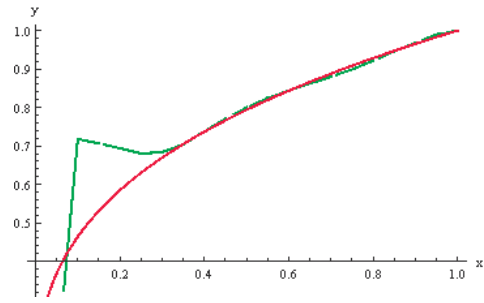


Figura 3.28: Solução obtida pelos métodos “Pesquisa Unidireccional”, Runge-Kutta, com *Piecewise Constant*.

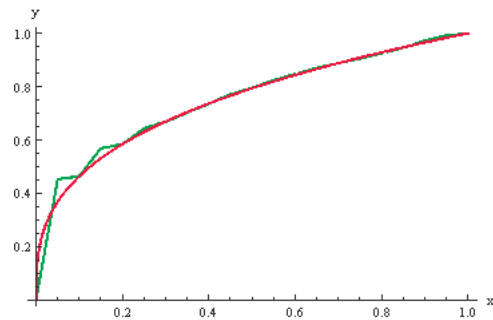


Figura 3.29: Solução obtida pelos métodos Pesquisa Directa 1, Euler, com *Piecewise Linear* que coincide com a solução obtida pelos métodos Pesquisa Directa 1, Euler, com *Piecewise Constant*.

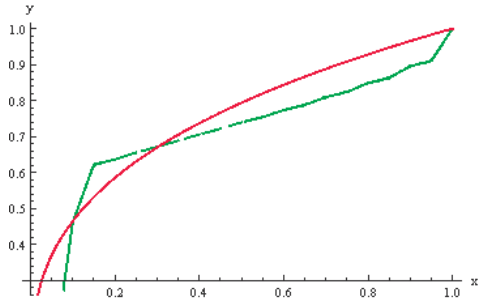


Figura 3.30: Solução obtida pelos métodos Pesquisa Directa 1, Runge-Kutta, com *Piecewise Linear*.

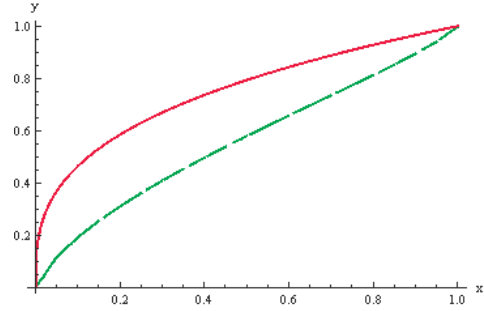


Figura 3.31: Solução obtida pelos métodos Pesquisa Directa 1, Runge-Kutta, com *Piecewise Constant*.

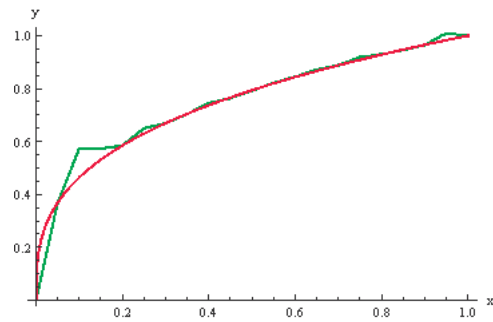


Figura 3.32: Solução obtida pelos métodos Pesquisa Directa 2, Euler, com *Piecewise Linear* que coincide com a solução obtida pelos métodos Pesquisa Directa 2, Euler, com *Piecewise Constant*.

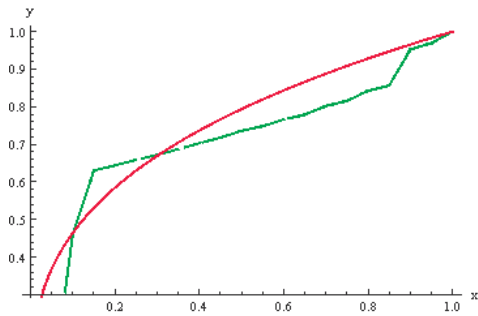


Figura 3.33: Solução obtida pelos métodos Pesquisa Directa 2, Runge-Kutta, com *Piecewise Linear*.

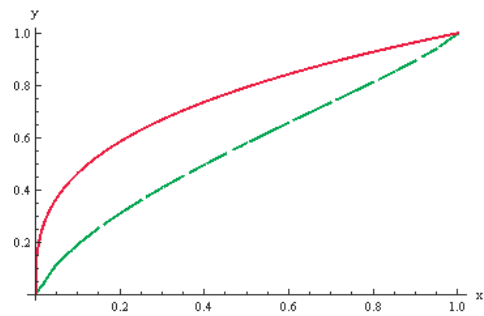


Figura 3.34: Solução obtida pelos métodos Pesquisa Directa 2, Runge-Kutta, com *Piecewise Constant*.

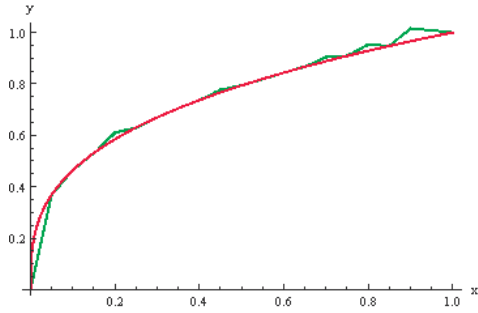


Figura 3.35: Solução obtida pelos métodos Pesquisa Aleatória, Euler, com *Piecewise Linear*.

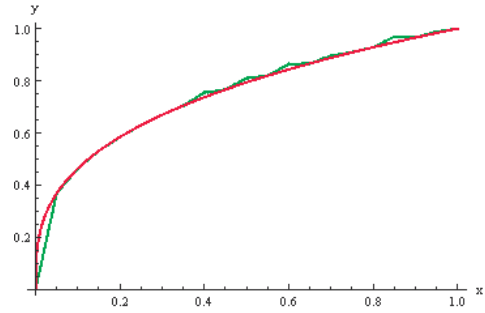


Figura 3.36: Solução obtida pelos métodos Pesquisa Aleatória, Euler, com *Piecewise Constant*.

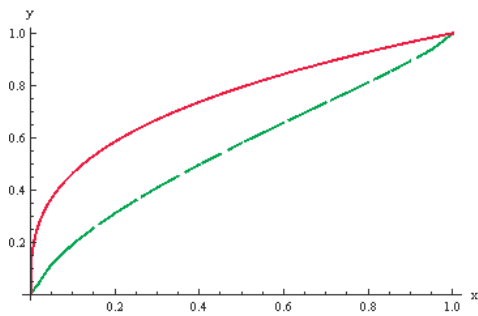


Figura 3.37: Solução obtida pelos métodos Pesquisa Aleatória, Runge-Kutta, com *Piecewise Linear*.

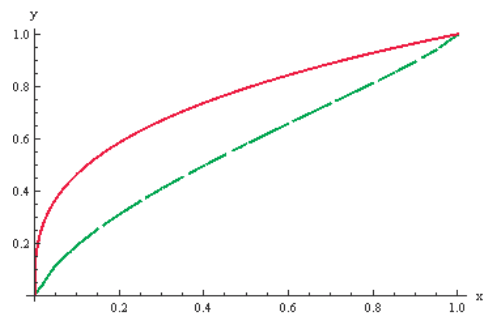


Figura 3.38: Solução obtida pelos métodos Pesquisa Aleatória, Runge-Kutta, com *Piecewise Constant*.

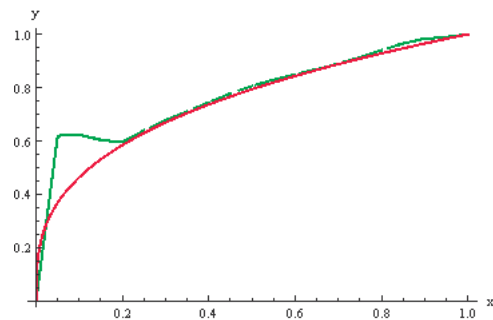


Figura 3.39: Solução obtida pelos métodos Gradientes Conjugados, Euler, com *Piecewise Linear* que coincide com a solução obtida pelos métodos Gradientes Conjugados, Euler, com *Piecewise Constant*.

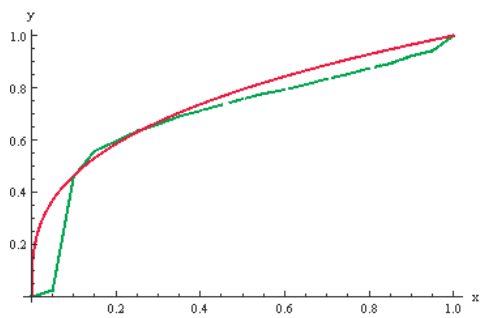


Figura 3.40: Solução obtida pelos métodos Gradientes Conjugados, Runge-Kutta, com *Piecewise Linear*.

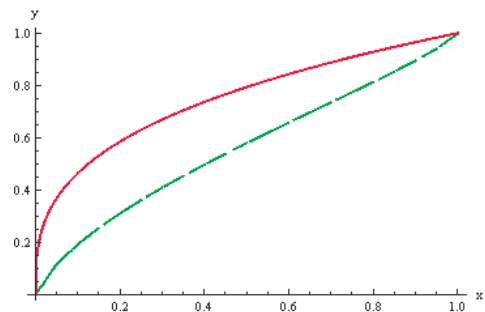


Figura 3.41: Solução obtida pelos métodos Gradientes Conjugados, Runge-Kutta, com *Piecewise Constant*.

3.5 Conclusão

Neste capítulo apresentaram-se os resultados obtidos pela implementação do algoritmo em estudo (*cf.* secção 3.2) aplicada a dois exemplos de problemas do Cálculo das Variações. Foram, ainda, referidos resultados fornecidos por outra ferramenta e por aplicação de outras estratégias de cálculo.

No primeiro exemplo estudado, o Problema da Braquistócrona, os resultados da implementação no Mathematica do algoritmo em estudo são melhores usando a Discretização de Guseinov do que a Discretização Usual. Além disto, estes valores melhoram à medida que é aumentado o número de intervalos em que se divide o domínio de integração. A melhor solução encontrada para o Problema da Braquistócrona (com a Discretização de Guseinov em 20 intervalos usando o método *Random Search*) tem valor de integral 8.18934.

Este valor, não sendo, obviamente, melhor que o valor óptimo, é melhor do que o valor calculado da curva criada a partir de pontos da curva óptima (8.19139).

Esta solução é, ainda, melhor do que a solução encontrada pela ferramenta OC (8.33608) e do que a solução obtida usando o algoritmo com estratégias evolucionistas (8.19365), considerando ambos 20 intervalos.

Tabela 3.5: Valores dos integrais das curvas construídas a partir das soluções encontradas pelos processos descritos.

Solução	Valor do Integral	Observações
Óptima	8.16470	
Pontos da curva óptima	8.19139	21 pontos com abcissas (sucessivas) equidistantes
Algoritmo em estudo	8.189344	20 intervalos Discretização de Guseinov Método <i>Random Search</i>
Algoritmo com estratégia evolucionista	8.19365	20 intervalos
Ferramenta OC	8.33608	20 intervalos Método <i>Conjugated Gradients</i> Método <i>Runge-Kutta 2</i> <i>Piecewise Linear</i>

No segundo exemplo estudado (problema proposto por Manià), observou-se que a interpretação dos resultados é menos óbvia devido à natureza do problema. Assim, ao contrário do problema anterior, a proximidade gráfica das soluções encontradas com a solução óptima (curva de equação $y = x^{\frac{1}{3}}$) parece não estar relacionada com aproximação obtida pelo in-

tegral para o valor óptimo (0). Por exemplo, dos resultados obtidos pela implementação do algoritmo (neste caso apenas usando a Discretização Usual), a melhor solução seria a encontrada pelo método *Simulated Annealing* com 8 intervalos cujo valor do integral é 0.327614. No entanto, comparando o gráfico da curva construída com essa solução (Figura 3.18, na página 63) com o gráfico da curva da solução obtida usando 15 intervalos de discretização e o método *Random Search* (Figura 3.22 na página 63), com valor de integral 1.15649, verifica-se que este segundo gráfico está mais próximo da curva óptima.

Desta forma, aumentando o número de intervalos em que se discretiza o problema, obtêm-se melhores soluções do ponto de vista gráfico, apesar do valor do integral não melhorar. Viu-se, ainda, que este facto é também observado nas curvas das funções definidas por ramos lineares a partir de pontos da solução óptima e nas soluções encontradas pela ferramenta OC. Com esta ferramenta, a melhor solução (usando o critério do valor do integral) foi obtida pelo método dos Gradientes Conjugados, Runge-Kutta, *Piecewise Constant* (Figura 3.41, página 71) com valor do integral 0.0326998 mas o gráfico da Figura 3.27 (página 68), cuja solução foi encontrada pelo método “Pesquisa Unidireccional”, Runge-Kutta *Piecewise Linear* mostra que esta solução está mais próxima da óptima, apesar de ter valor do integral 1.45193.

Assim, devido às particularidades já referidas deste exemplo, não se encontrou uma forma objectiva de comparar os resultados obtidos pelos vários métodos e ferramentas. No entanto, observando os gráficos, pode dizer-se que as soluções obtidas pelo algoritmo aqui em estudo são muito satisfatórias.

Com todos os dados apresentados neste capítulo, pode concluir-se que o algoritmo proposto fornece soluções melhores ou igualmente razoáveis quando comparadas com soluções obtidas por outros métodos ou ferramentas para problemas do Cálculo das Variações.

Capítulo 4

Conclusão

O Cálculo das Variações, como área da Matemática, tem já séculos de existência. No entanto, continua actual e a suscitar investigação a diferentes níveis. Actualmente procuram-se, não apenas métodos analíticos para resolver este tipo de problemas mas também ferramentas que forneçam as melhores soluções numéricas possíveis para problemas com interesse prático.

Com isto em vista, estudou-se e implementou-se computacionalmente, neste trabalho, um algoritmo proposto por Guseinov (3.2) que encontra soluções para problemas do Cálculo das Variações discretizando o seu domínio. No seu artigo([4]), este autor sugere uma discretização um pouco diferente da usual porque tem em conta a forma como a solução procurada se aproxima da solução óptima.

Um dos passos deste algoritmo consta na resolução de um sistema de equações não lineares, o que veio a revelar-se, em certa forma, problemático só por si. Qualquer ferramenta computacional de cálculo (*Mathematica*, *Maple*, *Matlab*, *Octave*, ...) resolve, algébrica e numericamente, sistemas de equações não lineares. No entanto, dependendo do número de intervalos da discretização, o sistema pode ter mais ou menos equações sendo, por isso, menos ou mais simples de resolver. Além disto, sucedeu que, muitas vezes, as soluções propostas pelas funções que resolvem sistemas de equações não lineares envolviam números complexos, o que não é aceitável para os problemas aqui estudados. Assim, optou-se por converter o sistema de equações não lineares num problema de programação não linear. Usando uma função do *Mathematica* para resolver este tipo de problemas, pôde escolher-se o método a aplicar.

Mais ainda, como por vezes alguns destes métodos não convergiam, propondo soluções com valores totalmente desadequados, elaborou-se um novo conjunto das restrições que indicam os limites superior e inferior (máximo e mínimo relativos da solução óptima no intervalo de integração) para cada elemento da solução.

Desta forma, a implementação do algoritmo foi testada com dois tipos de discretização em n ($n \in \mathbb{N}$) intervalos, usando diferentes métodos para resolver o problema de programação não linear. Ou seja, foram testadas as várias combinações destes elementos:

- Tipo de discretização: Usual e de Guseinov.
- Método para resolver problema de programação não linear: *Random Search*, *Nelder Mead*, *Differential Evolution* e *Simulated Annealing*.
- Número de intervalos em que se divide o domínio de integração: 3, 5, 8, 10, 15 e 20.
- Conjunto de restrições do problema de programação não linear: apenas Condições de Fronteira e Condições de Fronteira juntamente com intervalo ao qual pertencem todos os elementos da solução.

Os vários testes efectuados e relatados nesta dissertação mostram que o algoritmo fornece soluções muito razoáveis. Como seria de prever, as soluções melhoram à medida que se aumenta o número de intervalos da discretização, apresentando a Discretização de Guseinov melhores resultados que a Discretização Usual.

De referir, ainda, que aumentando o número de intervalos de discretização (para um valor superior a 20) o algoritmo encontra melhores soluções. No entanto, para estes valores, muitos dos métodos para resolver o problema de programação não linear deixam de convergir, não apresentando soluções aceitáveis.

Além disso, a implementação foi testada noutros exemplos como o Problema de Resistência Mínima de Newton (um problema clássico do Controlo Óptimo). Neste caso, as soluções encontradas não eram tão boas como as soluções obtidas para problemas do Cálculo das Variações, mesmo acrescentando a Equação ou Sistema de Controlo às restrições do problema de programação não linear. Por exemplo, o problema

$$\min_{y(x)} R[x, y(x), y'(x)] = \min_{y(x)} \int_0^1 \frac{x}{1 + (y'(x))^2} dx$$

em que $y(0) = 0$, $y(1) = 2$ e $y'(x) > 0$, tem como valor óptimo 0.08369157236. A melhor solução encontrada pela implementação do algoritmo no *Mathematica* tem como valor do integral 0.1 (Figura 4.1) e foi obtida usando 3 intervalos de discretização, o método *Random Search*, a Discretização de Guseinov e o segundo conjunto de restrições (neste problema, considerando apenas as Condições de Fronteira e a Equação de Controlo – $y'(x) > 0$ –, os métodos não convergem). Para $n = 20$, a melhor solução foi fornecida pelo método *Nelder Mead*, com Discretização Usual, tendo valor do integral 0.390947 (Figura 4.2).

Alterando as Condições de Fronteira deste problema para, por exemplo, $y(0) = 0$ e $y(2) = 1$ ou $y(0) = 0$ e $y(1) = 1$, os resultados pioram. Por isto se considerou que este algoritmo não fornece soluções tão boas para problemas que não sejam do Cálculo das Variações.

Para averiguar se as aproximações obtidas pela implementação do algoritmo são razoáveis, testaram-se outras ferramentas de cálculo específicas para problemas do Cálculo das Variações e Controlo Óptimo com os mesmos exemplos. Destes testes e a partir de resultados apresentados na bibliografia consultada conclui-se que a implementação do algoritmo proposto

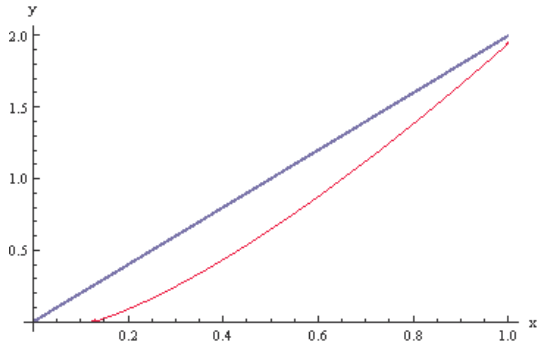


Figura 4.1: Solução óptima do Problema de Resistência Mínima (vermelho) e curva definida pela solução encontrada usando a Discretização de Guseinov, $n = 3$ e método *Random Search* (azul).

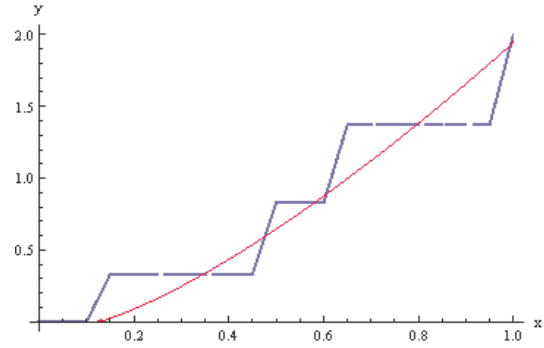


Figura 4.2: Solução óptima do Problema de Resistência Mínima (vermelho) e curva definida pela solução encontrada usando a Discretização Usual, $n = 20$ e método *Nelder Mead* (azul).

apresenta muito boas soluções para o Problema da Braquistócrona (melhores do que a ferramenta OC (2.3) e do que o algoritmo evolucionista (2.4)) e boas soluções para o problemático Exemplo de Manià (problema do Cálculo das Variações que exhibe o Fenómeno de Lavrentiev).

Em termos de resultados, ambas as ferramentas computacionais de cálculo apresentam aproximações muito satisfatórias mesmo para exemplos problemáticos. Mas a utilização de cada ferramenta é diferente tanto na forma como é aplicado o problema como na apresentação das soluções encontradas. Comparando a implementação efectuada com o programa OC, cada ferramenta tem vantagens e desvantagens em comparação com a outra.

Vantagens da implementação do algoritmo (3.2):

- A introdução dos dados é muito simples e imediata desde que o problema do Cálculo das Variações esteja na forma de Lagrange (1.1), a forma mais usual;
- Apresenta valor do funcional contínuo e discreto;
- Testa os diferentes métodos e escolhe o melhor resultado.

A única desvantagem desta implementação diz respeito ao valor de n , como já foi referido. Para valores grandes de n a resolução do sistema de equações não lineares (problema de PNL) pode ser complicada e, por isso, não encontrar boas soluções para o problema.

Também a ferramenta OC (2.3) tem vantagens e desvantagens. A mais valia deste *software* é a facilidade com que são testados os diferentes métodos que o programa permite usar. Mas, por outro lado, a criação da aplicação específica de um problema não é imediata:

- O problema deve estar na forma de Mayer (2.4) e, por isso, pode ser necessário efectuar algum tipo de conversão.

- É necessário usar código *C++* (para descrever o funcional).

O problema deve ser introduzido num ficheiro do tipo *cpp* usando um dos exemplos fornecidos pelos autores da ferramenta. Segue-se a compilação do novo código. Este processo, não sendo complicado, não é imediato para muitos utilizadores.

Outra desvantagem desta ferramenta é o facto de não apresentar o valor do funcional da solução encontrada. Além disto, as soluções encontradas dependem do “Limite de Iterações” escolhidos (alguns métodos apresentam soluções para determinado número de iterações e terminam com o aviso “Use another method” com outro número de iterações). Por isto, pode ser necessário efectuar vários testes.

Acerca do algoritmo proposto e da sua implementação pode concluir-se que apresenta soluções muito satisfatórias, sendo de fácil utilização e funciona melhor em problemas do Cálculo das Variações com significado físico (para ser possível usar a Discretização de Guseinov) do que em exemplos apenas teóricos.

Este trabalho deixa em aberto algumas questões que podem ser resolvidas ou abordadas em trabalhos futuros.

Foram testados apenas exemplos com solução óptima conhecida mas, na realidade, interessa aplicar ferramentas de cálculo a problemas com solução desconhecida e, por isso, antes de aplicar o algoritmo aqui estudado, pode, por exemplo, averiguar-se se um problema tem solução ou se uma solução encontrada minimiza ou maximiza o problema em estudo.

Além disto, pode tentar-se melhorar ou ajustar o algoritmo ou a implementação de modo que seja aplicável a problemas de Controlo Óptimo.

Apêndice A

Cálculos em *Maple*

A.1 Capítulo 1, Exemplo 2

```

> restart;
> f := (x,y,v) -> y*sqrt(1+v^2);
      
$$f := (x, y, v) \mapsto y\sqrt{1+v^2}$$

> fy := subs(v=diff(y(x),x),diff(f(x,y,v),y));
      
$$fy := \sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}$$

> fv := subs({y=y(x),v=diff(y(x),x)},diff(f(x,y,v),v));
      
$$fv := \frac{y(x) \frac{d}{dx}y(x)}{\sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}}$$

> EL := diff(fv,x)=fy;

$$EL := \frac{\left(\frac{d}{dx}y(x)\right)^2}{\sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}} - \frac{y(x) \left(\frac{d}{dx}y(x)\right)^2 \frac{d^2}{dx^2}y(x)}{\left(1 + \left(\frac{d}{dx}y(x)\right)^2\right)^{3/2}} + \frac{y(x) \frac{d^2}{dx^2}y(x)}{\sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}} = \sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}$$

> sol := dsolve(EL,y(x));

$$sol := y(x) = 1/2\_C1 \left( \left( e^{\frac{x}{\_C1}} \right)^{-2} \left( e^{\frac{-C2}{\_C1}} \right)^{-2} + 1 \right) e^{\frac{x}{\_C1}} e^{\frac{-C2}{\_C1}},$$


$$y(x) = 1/2\_C1 \left( \left( e^{\frac{x}{\_C1}} \right)^2 \left( e^{\frac{C2}{\_C1}} \right)^2 + 1 \right) \left( e^{\frac{x}{\_C1}} \right)^{-1} \left( e^{\frac{C2}{\_C1}} \right)^{-1}$$

> y1 := unapply(rhs(sol[1]),x):
> y2 := unapply(rhs(sol[2]),x):
> y1(x);

$$1/2\_C1 \left( \left( e^{\frac{x}{\_C1}} \right)^{-2} \left( e^{\frac{-C2}{\_C1}} \right)^{-2} + 1 \right) e^{\frac{x}{\_C1}} e^{\frac{-C2}{\_C1}}$$

> y2(x);

```

$$\frac{1}{2} C1 \left(\left(e^{\frac{x}{C1}} \right)^2 \left(e^{\frac{C2}{C1}} \right)^2 + 1 \right) \left(e^{\frac{x}{C1}} \right)^{-1} \left(e^{\frac{C2}{C1}} \right)^{-1}$$

```

> simplify(y1(x)-y2(x)); # afinal só há uma extremal; y1 = y2
0
> const := fsolve({y1(-2)=3,y1(1)=2},{_C1,_C2});
const := {_C1 = 1.687122717, _C2 = 0.01223389066}
> extremal := subs(const,y1(x));
extremal := 0.8435613585  $\left( \frac{1}{(e^{0.5927251112 x})^2 (e^{0.007251334202})^2} + 1 \right) e^{0.5927251112 x} e^{0.007251334202}$ 
> valorOptimo := evalf(int(extremal*sqrt(1+diff(extremal,x)^2),x=-2..1));
valorOptimo := 7.430431404

```

A.2 Capítulo 1, Exemplo 1 usando *VariationalCalculus*

Resolução do problema do Exemplo 1 (página 12) usando a ferramenta *VariationalCalculus* do *Maple* 10.

```
> restart;  
> with(VariationalCalculus);  
      [ConjugateEquation, Convex, EulerLagrange, Jacobi, Weierstrass]  
> f := (1+(diff(y(x), x))^2)^(1/2);
```

$$f := \sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}$$

```
> expre := EulerLagrange(f, x, y(x))[1];
```

$$\text{expre} := \frac{\left(\frac{d}{dx}y(x)\right)^2 \frac{d^2}{dx^2}y(x)}{\left(1 + \left(\frac{d}{dx}y(x)\right)^2\right)^{3/2}} - \frac{\frac{d^2}{dx^2}y(x)}{\sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}}$$

```
> edo := expre = 0;
```

$$\text{edo} := \frac{\left(\frac{d}{dx}y(x)\right)^2 \frac{d^2}{dx^2}y(x)}{\left(1 + \left(\frac{d}{dx}y(x)\right)^2\right)^{3/2}} - \frac{\frac{d^2}{dx^2}y(x)}{\sqrt{1 + \left(\frac{d}{dx}y(x)\right)^2}} = 0$$

```
> cfronteira := y(-1) = 2, y(5) = 4;
```

$$\text{cfronteira} := y(-1) = 2, y(5) = 4$$

```
> dsolve(edo, cfronteira);
```

$$y(x) = 1/3x + 7/3$$

Apêndice B

Implementação no *Mathematica* 6.0

B.1 Problema da Braquistócrona

Problema da Braquistócrona com “Discretização de Guseinov”, para $n=5$ apenas com as Condições de Fronteira

```
(*reset*)
```

```
ClearAll[“*”]
```

```
(*input*)
```

```
(*braquistócrona*)
```

```
f[x_, y_, z_] := Sqrt[1 + z^2]/Sqrt[ya - y]
```

```
L[s_, t_, u_, v_, w_] := Sqrt[1 + w^2]/(Sqrt[ya - u] + Sqrt[ya - v])
```

```
(*de Guseinov*)
```

```
xa = 0;
```

```
xb = 10;
```

```
ya = 10;
```

```
yb = 0;
```

```
n = 5;
```

```
(*somatório*)
```

```
funcionald := Function[{X, Y},
```

```
Sum[L[X[[i]], X[[i + 1]], Y[[i]], Y[[i + 1]], (Y[[i + 1]] - Y[[i]])/Δx]Δx, {i, 1, n}]];
```

```
(*derivadas parciais*)
```

```
Lu[s_, t_, u_, v_, w_] = D[L[s, t, u, v, w], u]
```

```
Lv[s_, t_, u_, v_, w_] = D[L[s, t, u, v, w], v]
```

```
Lw[s_, t_, u_, v_, w_] = D[L[s, t, u, v, w], w]
```

```

$$\frac{\sqrt{1+w^2}}{2\sqrt{10-u}(\sqrt{10-u}+\sqrt{10-v})^2}$$

```

$$\frac{\frac{\sqrt{1+w^2}}{2(\sqrt{10-u}+\sqrt{10-v})^2\sqrt{10-v}}}{\frac{w}{(\sqrt{10-u}+\sqrt{10-v})\sqrt{1+w^2}}}$$

(*vector com extremos dos n intervalos em que se particiona o domínio*)

$$\Delta \mathbf{x} = (\mathbf{x}_b - \mathbf{x}_a)/n;$$

$$\text{abcissas} = \text{Range}[\mathbf{x}_a, \mathbf{x}_b, \Delta \mathbf{x}]$$

$$\{0, 2, 4, 6, 8, 10\}$$

(*equações não lineares*)

Equacao =

$$(\text{Lu}[X[i+1], X[i+2], Y[i+1], Y[i+2], (Y[i+2] - Y[i+1])/\Delta \mathbf{x}]$$

$$+\text{Lv}[X[i], X[i+1], Y[i], Y[i+1], (Y[i+1] - Y[i])/\Delta \mathbf{x}]-$$

$$(\text{Lw}[X[i+1], X[i+2], Y[i+1], Y[i+2], (Y[i+2] - Y[i+1])/\Delta \mathbf{x}]-$$

$$\text{Lw}[X[i], X[i+1], Y[i], Y[i+1], (Y[i+1] - Y[i])/\Delta \mathbf{x}])/\Delta \mathbf{x} == 0)$$

$$\frac{\frac{\sqrt{1+\frac{1}{4}(-Y[i]+Y[1+i])^2}}{2(\sqrt{10-Y[i]}+\sqrt{10-Y[1+i]})^2\sqrt{10-Y[1+i]}}}{\frac{\sqrt{1+\frac{1}{4}(-Y[1+i]+Y[2+i])^2}}{2\sqrt{10-Y[1+i]}(\sqrt{10-Y[1+i]}+\sqrt{10-Y[2+i]})^2}} + \frac{1}{2} \left(\frac{\frac{-Y[i]+Y[1+i]}{2(\sqrt{10-Y[i]}+\sqrt{10-Y[1+i]})\sqrt{1+\frac{1}{4}(-Y[i]+Y[1+i])^2}}}{\frac{-Y[1+i]+Y[2+i]}{2(\sqrt{10-Y[1+i]}+\sqrt{10-Y[2+i]})\sqrt{1+\frac{1}{4}(-Y[1+i]+Y[2+i])^2}}} - \right) = 0$$

(*condições de fronteira*)

$$\mathbf{C1} = Y[1] == \mathbf{ya};$$

$$\mathbf{C2} = Y[n+1] == \mathbf{yb};$$

$$X[\mathbf{i}]:= \text{abcissas}[[i]]$$

$$\text{Sistema} = \text{Flatten}\{\{\text{Table}[\text{Equacao}, \{\mathbf{i}, 1, n-1\}], \mathbf{C1}, \mathbf{C2}\}$$

$$\left\{ \frac{\frac{\sqrt{1+\frac{1}{4}(-Y[1]+Y[2])^2}}{2(\sqrt{10-Y[1]}+\sqrt{10-Y[2]})^2\sqrt{10-Y[2]}}}{\frac{\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}{2\sqrt{10-Y[2]}(\sqrt{10-Y[2]}+\sqrt{10-Y[3]})^2}} + \frac{1}{2} \left(\frac{\frac{-Y[1]+Y[2]}{2(\sqrt{10-Y[1]}+\sqrt{10-Y[2]})\sqrt{1+\frac{1}{4}(-Y[1]+Y[2])^2}}}{\frac{-Y[2]+Y[3]}{2(\sqrt{10-Y[2]}+\sqrt{10-Y[3]})\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}} - \right) == 0, \right. \\ \left. \frac{\frac{\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}{2(\sqrt{10-Y[2]}+\sqrt{10-Y[3]})^2\sqrt{10-Y[3]}}}{\frac{\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}}{2\sqrt{10-Y[3]}(\sqrt{10-Y[3]}+\sqrt{10-Y[4]})^2}} + \frac{1}{2} \left(\frac{\frac{-Y[2]+Y[3]}{2(\sqrt{10-Y[2]}+\sqrt{10-Y[3]})\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}}{\frac{-Y[3]+Y[4]}{2(\sqrt{10-Y[3]}+\sqrt{10-Y[4]})\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}}} - \right) == 0, \right.$$

$$\begin{aligned}
& \frac{\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}}{2\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)^2\sqrt{10-Y[4]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}}{2\sqrt{10-Y[4]}\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)^2} + \\
& \frac{1}{2} \left(\frac{-Y[3]+Y[4]}{2\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}} - \right. \\
& \left. \frac{-Y[4]+Y[5]}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}} \right) == 0, \\
& \frac{\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)^2\sqrt{10-Y[5]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[5]+Y[6])^2}}{2\sqrt{10-Y[5]}\left(\sqrt{10-Y[5]}+\sqrt{10-Y[6]}\right)^2} + \\
& \frac{1}{2} \left(\frac{-Y[4]+Y[5]}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}} - \right. \\
& \left. \frac{-Y[5]+Y[6]}{2\left(\sqrt{10-Y[5]}+\sqrt{10-Y[6]}\right)\sqrt{1+\frac{1}{4}(-Y[5]+Y[6])^2}} \right) == 0, Y[1] == 10, Y[6] == 0 \}
\end{aligned}$$

Obj = Total[Table[(Sistema[[i, 1]] - Sistema[[i, 2]])^2, {i, 1, n + 1}]]

$$\begin{aligned}
& (-10 + Y[1])^2 + \left(\frac{\sqrt{1+\frac{1}{4}(-Y[1]+Y[2])^2}}{2\left(\sqrt{10-Y[1]}+\sqrt{10-Y[2]}\right)^2\sqrt{10-Y[2]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}{2\sqrt{10-Y[2]}\left(\sqrt{10-Y[2]}+\sqrt{10-Y[3]}\right)^2} + \right. \\
& \left. \frac{1}{2} \left(\frac{-Y[1]+Y[2]}{2\left(\sqrt{10-Y[1]}+\sqrt{10-Y[2]}\right)\sqrt{1+\frac{1}{4}(-Y[1]+Y[2])^2}} - \frac{-Y[2]+Y[3]}{2\left(\sqrt{10-Y[2]}+\sqrt{10-Y[3]}\right)\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}} \right) \right)^2 + \\
& \left(\frac{\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}}{2\left(\sqrt{10-Y[2]}+\sqrt{10-Y[3]}\right)^2\sqrt{10-Y[3]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}}{2\sqrt{10-Y[3]}\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)^2} + \right. \\
& \left. \frac{1}{2} \left(\frac{-Y[2]+Y[3]}{2\left(\sqrt{10-Y[2]}+\sqrt{10-Y[3]}\right)\sqrt{1+\frac{1}{4}(-Y[2]+Y[3])^2}} - \frac{-Y[3]+Y[4]}{2\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}} \right) \right)^2 + \\
& \frac{\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}}{2\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)^2\sqrt{10-Y[4]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}}{2\sqrt{10-Y[4]}\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)^2} + \\
& \frac{1}{2} \left(\frac{-Y[3]+Y[4]}{2\left(\sqrt{10-Y[3]}+\sqrt{10-Y[4]}\right)\sqrt{1+\frac{1}{4}(-Y[3]+Y[4])^2}} - \frac{-Y[4]+Y[5]}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}} \right) \right)^2 + \\
& Y[6]^2 + \left(\frac{\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)^2\sqrt{10-Y[5]}} + \frac{\sqrt{1+\frac{1}{4}(-Y[5]+Y[6])^2}}{2\sqrt{10-Y[5]}\left(\sqrt{10-Y[5]}+\sqrt{10-Y[6]}\right)^2} + \right. \\
& \left. \frac{1}{2} \left(\frac{-Y[4]+Y[5]}{2\left(\sqrt{10-Y[4]}+\sqrt{10-Y[5]}\right)\sqrt{1+\frac{1}{4}(-Y[4]+Y[5])^2}} - \frac{-Y[5]+Y[6]}{2\left(\sqrt{10-Y[5]}+\sqrt{10-Y[6]}\right)\sqrt{1+\frac{1}{4}(-Y[5]+Y[6])^2}} \right) \right)^2
\end{aligned}$$

fObj = Function[Y, Obj];

ym = Max[Abs[ya], Abs[yb]];

vars = Table[Y[i], {i, 1, n + 1}];

cons = {Y[1] == ya, Y[n + 1] == yb};

(*Método “Random Search”*)

sol1 = NMinimize[{fObj[vars], cons}, vars, Method → “RandomSearch”]//Timing

```
{1.125, {5.837975143923809×10-34, {Y[1] → 10., Y[2] → 5.2357, Y[3] → 3.29408, Y[4] → 1.87594, Y[5] → 0.804153, Y[6] → 0}}}}
```

```
solnum1 = Table[Y[i], {i, 1, n + 1}]/.sol1[[2, 2]]
```

```
valorfunc1 = funcionald[abscissas, solnum1]
```

```
{10., 5.2357, 3.29408, 1.87594, 0.804153, 0}
```

```
4.13574
```

```
(*Método “NelderMead”*)
```

```
sol2 = NMinimize[{fObj[vars], cons}, vars, Method → “NelderMead”]//Timing
```

```
{0.156, {2.035015820143544×10-33, {Y[1] → 10., Y[2] → 5.2357, Y[3] → 3.29408, Y[4] → 1.87594, Y[5] → 0.804153, Y[6] → 0}}}}
```

```
solnum2 = Table[Y[i], {i, 1, n + 1}]/.sol2[[2, 2]]
```

```
valorfunc2 = funcionald[abscissas, solnum2]
```

```
{10., 5.2357, 3.29408, 1.87594, 0.804153, 0}
```

```
4.13574
```

```
solnum1===solnum2
```

```
False
```

```
(*Método “Differential Evolution”*)
```

```
sol3 = NMinimize[{fObj[vars], cons}, vars, Method → “DifferentialEvolution”]//Timing
```

```
NMinimize::nrnum : The function value  $-0.0252334 + 0.00118166i$  is not a real number at  
{Y[1], Y[2], Y[3], Y[4], Y[5], Y[6]} = {10., 15.0812, 2.98809, -1.92832, -0.319437, 0}. >>
```

```
{0.047, {0.00172661, {Y[1] → 10., Y[2] → 0.483033, Y[3] → -0.607396, Y[4] → -0.951742, Y[5] → -0.650568, Y[6] → 0}}}}
```

```
solnum3 = Table[Y[i], {i, 1, n + 1}]/.sol3[[2, 2]]
```

```
valorfunc3 = funcionald[abscissas, solnum3]
```

```
{10., 0.483033, -0.607396, -0.951742, -0.650568, 0}
```

```
4.45562
```

```
solnum1===solnum3
```

```
solnum2===solnum3
```

```
False
```

```
False
```

(*Método “Simulated Annealing”*)

```
sol4 = NMinimize[{fObj[vars], cons}, vars, Method → “SimulatedAnnealing”]//Timing
{0.172, {2.035015820143544×10-33, {Y[1] → 10., Y[2] → 5.2357, Y[3] → 3.29408, Y[4] → 1.87594, Y[5] → 0.804153, Y[6] → 0}}}}
```

```
solnum4 = Table[Y[i], {i, 1, n + 1}]/.sol4[[2, 2]]
valorfunc4 = funcionald[abscissas, solnum4]
{10., 5.2357, 3.29408, 1.87594, 0.804153, 0}
4.13574
```

```
solnum1===solnum4
```

```
solnum2===solnum4
```

```
solnum3===solnum4
```

```
False
```

```
True
```

```
False
```

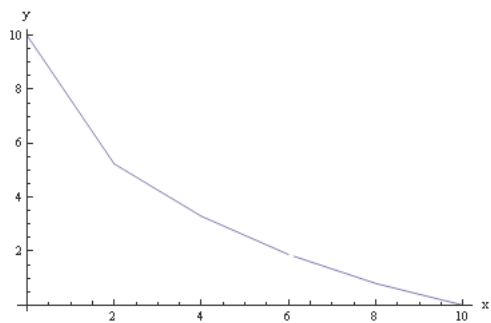
(*método random search*)

```
y1[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[
{{X[i], solnum1[[i]]}, {X[i + 1], solnum1[[i + 1]]}], x], x], X[i] <= x < X[i + 1]], {i, 1, n}]]

$$\begin{cases} 10. - 2.38215x & 0 \leq x < 2 \\ 7.17732 - 0.97081x & 2 \leq x < 4 \\ 6.13035 - 0.709069x & 4 \leq x < 6 \\ 5.09131 - 0.535895x & 6 \leq x < 8 \\ 4.02076 - 0.402076x & 8 \leq x < 10 \end{cases}$$

```

```
Plot[y1[x], {x, xa, xb}, AxesLabel → {“x”, “y”}]
```



```
iy1 = Integrate[f[x, y1[x], D[y1[x], x]], {x, xa, xb}]
```

```
8.27147
```

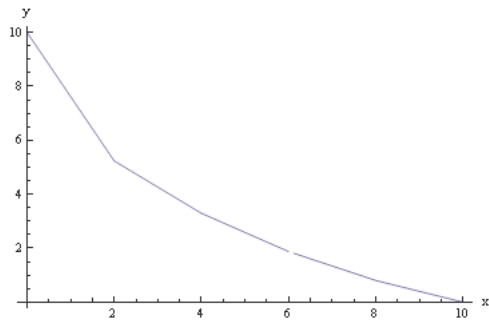
(*método nelder mead*)

y2[x_] =

```
Piecewise[Table[{Collect[InterpolatingPolynomial[
{{X[i], solnum2[[i]]}, {X[i + 1], solnum2[[i + 1]]}], x], x], X[i] <= x < X[i + 1]}, {i, 1, n}]]
```

$$\begin{cases} 10. - 2.38215x & 0 \leq x < 2 \\ 7.17732 - 0.97081x & 2 \leq x < 4 \\ 6.13035 - 0.709069x & 4 \leq x < 6 \\ 5.09131 - 0.535895x & 6 \leq x < 8 \\ 4.02076 - 0.402076x & 8 \leq x < 10 \end{cases}$$

Plot[y2[x], {x, xa, xb}, AxesLabel → {"x", "y"}]



iy2 = Integrate[f[x, y2[x], D[y2[x], x]], {x, xa, xb}]

8.27147

(*método differential evolution*)

y3[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[

```
{{X[i], solnum3[[i]]}, {X[i + 1], solnum3[[i + 1]]}], x], x], X[i] <= x < X[i + 1]}, {i, 1, n}]]
```

$$\begin{cases} 10. - 4.75848x & 0 \leq x < 2 \\ 1.57346 - 0.545215x & 2 \leq x < 4 \\ 0.0812951 - 0.172173x & 4 \leq x < 6 \\ -1.85526 + 0.150587x & 6 \leq x < 8 \\ -3.25284 + 0.325284x & 8 \leq x < 10 \end{cases}$$

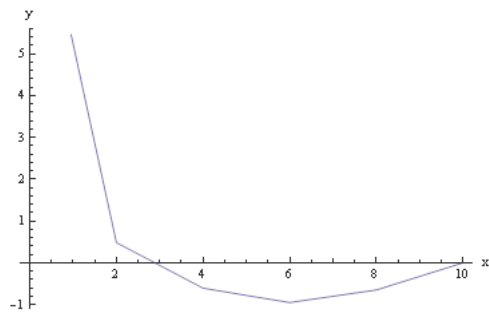
Plot[y3[x], {x, xa, xb}, AxesLabel → {"x", "y"}]

iy3 = Integrate[f[x, y3[x], D[y3[x], x]], {x, xa, xb}]

8.91123

(*método simulated annealing*)

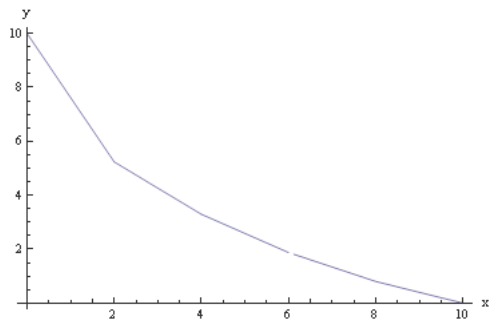
y4[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[



```
{ {X[i], solnum4[[i]]}, {X[i + 1], solnum4[[i + 1]]}}, x], x], X[i] <= x < X[i + 1]], {i, 1, n}]]
```

$$\begin{cases} 10. - 2.38215x & 0 \leq x < 2 \\ 7.17732 - 0.97081x & 2 \leq x < 4 \\ 6.13035 - 0.709069x & 4 \leq x < 6 \\ 5.09131 - 0.535895x & 6 \leq x < 8 \\ 4.02076 - 0.402076x & 8 \leq x < 10 \end{cases}$$

```
Plot[y4[x], {x, xa, xb}, AxesLabel -> {"x", "y"}]
```



```
iy4 = Integrate[f[x, y4[x], D[y4[x], x]], {x, xa, xb}]
```

```
8.27147
```

qual o método que apresenta a melhor solução do sistema?
função objectivo do Problema de Programação Não Linear

```
ValorfObjMin = Min[sol1[[2, 1]], sol2[[2, 1]], sol3[[2, 1]], sol4[[2, 1]]]
```

```
MetfObjMin = Switch[ValorfObjMin, sol1[[2, 1]], Print["Random Search"],
```

```
sol2[[2, 1]], Print["Nelder Mead"], sol3[[2, 1]],
```

```
Print["Differential Evolution"], sol4[[2, 1]],
```

```
Print["Simulated Annealing"]]
```

```
SolfObjMin = Switch[ValorfObjMin, sol1[[2, 1]], solnum1, sol2[[2, 1]],
```

```
solnum2, sol3[[2, 1]], solnum3, sol4[[2, 1]], solnum4];
```

$5.837975143923809 \times 10^{-34}$

Random Search

qual o método que apresenta a melhor solução do problema inicial?
valor do funcional discreto (somatório)

```
ValorFuncionalMin = Min[valorfunc1, valorfunc2, valorfunc3, valorfunc4]  
SolFuncionalMin = Switch[ValorFuncionalMin, valorfunc1,  
Print["Random Search"], valorfunc2, Print["Nelder Mead"],  
valorfunc3, Print["Differential Evolution"], valorfunc4,  
Print["Simulated Annealing"]]  
SolNumMin = Switch[ValorFuncionalMin, valorfunc1, solnum1, valorfunc2,  
solnum2, valorfunc3, solnum3, valorfunc4, solnum4];
```

4.13574

Random Search

qual o método que apresenta a melhor solução do problema inicial?
valor do funcional continuo

```
ValorIntegralMin = Min[iy1, iy2, iy3, iy4]  
SolIntegralMin = Switch[ValorIntegralMin, iy1, Print["Random Search"],  
iy2, Print["Nelder Mead"], iy3, Print["Differential Evolution"],  
iy4, Print["Simulated Annealing"]]  
FuncIntMin[x_] = Switch[ValorIntegralMin, iy1, y1[x], iy2, y2[x],  
iy3, y3[x], iy4, y4[x]];
```

8.27147

Random Search

gráfico da aproximação de integral mínimo

```
gmin = Plot[FuncIntMin[x], {x, xa, xb}, AxesLabel -> {"x", "y"}, PlotStyle -> Thick];
```

solução óptima para problema (0,10) -> (10,0):

função paramétrica

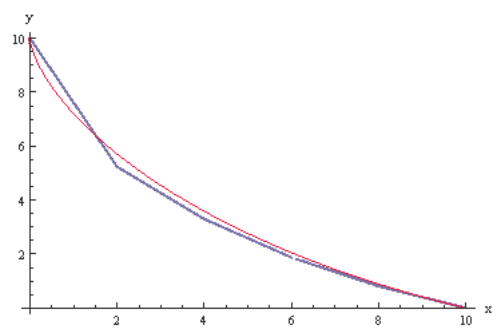
representação gráfica

$x[\theta] := 5.729170375\theta - 5.729170375\sin[\theta];$

$y[\theta] := 4.270829625 + 5.729170375\cos[\theta];$

```
gopt = ParametricPlot[{x[θ], y[θ]}, {θ, 0, 2.412011144},  
AxesLabel -> {"x", "y"}, ColorFunction -> Function[{x, y}, Red]];
```


Show[gmin, gopt]



B.2 Problema com Fenómeno de Lavrentiev

Problema com Fenómeno de Lavrentiev, usando a “Discretização Usual”, para $n = 15$ com as Condições de Fronteira e as condições que garantem que o valor de cada ponto da solução está entre 0 e o $\max \{|\alpha|, |\beta|\}$.

(*reset*)

ClearAll[“*”]

(*input*)

(*fenomenolavrentiev – exemplodemania*)

$f[x_ , y_ , z_] := (y^3 - x)^2 z^6$

$L[s_ , t_ , u_ , v_ , w_] := (v^3 - t)^2 w^6$ (*usual*)

xa = 0;

xb = 1;

ya = 0;

yb = 1;

n = 15;

(*somatorio*)

funcionald:=Function[{X, Y}, Sum[L[X[[i]], X[[i + 1]], Y[[i]], Y[[i + 1]],

(Y[[i + 1]] - Y[[i]])/Δx]Δx, {i, 1, n}]];

(*derivadas parciais*)

$Lu[s_ , t_ , u_ , v_ , w_] = D[L[s, t, u, v, w], u]$

$Lv[s_ , t_ , u_ , v_ , w_] = D[L[s, t, u, v, w], v]$

$Lw[s_ , t_ , u_ , v_ , w_] = D[L[s, t, u, v, w], w]$

0

$6v^2 (-t + v^3) w^6$

$6 (-t + v^3)^2 w^5$

(*vector com extremos dos n intervalos em que se particiona o domínio*)

$\Delta x = (xb - xa)/n$;

abscissas = Range[xa, xb, Δx]

$\{0, \frac{1}{15}, \frac{2}{15}, \frac{1}{5}, \frac{4}{15}, \frac{1}{3}, \frac{2}{5}, \frac{7}{15}, \frac{8}{15}, \frac{3}{5}, \frac{2}{3}, \frac{11}{15}, \frac{4}{5}, \frac{13}{15}, \frac{14}{15}, 1\}$

(*equações não lineares*)

Equacao =

$(Lu[X[i + 1], X[i + 2], Y[i + 1], Y[i + 2], (Y[i + 2] - Y[i + 1])/Δx]$

$+Lv[X[i], X[i + 1], Y[i], Y[i + 1], (Y[i + 1] - Y[i])/Δx] -$

(Lw[X[i + 1], X[i + 2], Y[i + 1], Y[i + 2], (Y[i + 2] - Y[i + 1])/Δx] - Lw[X[i], X[i + 1], Y[i], Y[i + 1], (Y[i + 1] - Y[i])/Δx])/Δx == 0)

$$68343750Y[1 + i]^2 (-Y[i] + Y[1 + i])^6 (-X[1 + i] + Y[1 + i]^3) - 15 \left(-4556250 (-Y[i] + Y[1 + i])^5 (-X[1 + i] + Y[1 + i]^3)^2 + 4556250 (-Y[1 + i] + Y[2 + i])^5 (-X[2 + i] + Y[2 + i]^3)^2 \right) == 0$$

(*condições de fronteira*)

C1 = Y[1] == ya;

C2 = Y[n + 1] == yb;

X[i_]:=abcissas[[i]]

Sistema = Flatten[{Table[Equacao, {i, 1, n - 1}], C1, C2}]

$$\left\{ 68343750Y[2]^2 (-Y[1] + Y[2])^6 \left(-\frac{1}{15} + Y[2]^3 \right) - 15 \left(-4556250 (-Y[1] + Y[2])^5 \left(-\frac{1}{15} + Y[2]^3 \right)^2 + 4556250 (-Y[2] + Y[3])^5 \left(-\frac{2}{15} + Y[3]^3 \right)^2 \right) == 0, \right.$$

$$68343750Y[3]^2 (-Y[2] + Y[3])^6 \left(-\frac{2}{15} + Y[3]^3 \right) - 15 \left(-4556250 (-Y[2] + Y[3])^5 \left(-\frac{2}{15} + Y[3]^3 \right)^2 + 4556250 (-Y[3] + Y[4])^5 \left(-\frac{1}{5} + Y[4]^3 \right)^2 \right) == 0,$$

$$68343750Y[4]^2 (-Y[3] + Y[4])^6 \left(-\frac{1}{5} + Y[4]^3 \right) - 15 \left(-4556250 (-Y[3] + Y[4])^5 \left(-\frac{1}{5} + Y[4]^3 \right)^2 + 4556250 (-Y[4] + Y[5])^5 \left(-\frac{4}{15} + Y[5]^3 \right)^2 \right) == 0,$$

$$68343750Y[5]^2 (-Y[4] + Y[5])^6 \left(-\frac{4}{15} + Y[5]^3 \right) - 15 \left(-4556250 (-Y[4] + Y[5])^5 \left(-\frac{4}{15} + Y[5]^3 \right)^2 + 4556250 (-Y[5] + Y[6])^5 \left(-\frac{1}{3} + Y[6]^3 \right)^2 \right) == 0,$$

$$68343750Y[6]^2 (-Y[5] + Y[6])^6 \left(-\frac{1}{3} + Y[6]^3 \right) - 15 \left(-4556250 (-Y[5] + Y[6])^5 \left(-\frac{1}{3} + Y[6]^3 \right)^2 + 4556250 (-Y[6] + Y[7])^5 \left(-\frac{2}{5} + Y[7]^3 \right)^2 \right) == 0,$$

$$68343750Y[7]^2 (-Y[6] + Y[7])^6 \left(-\frac{2}{5} + Y[7]^3 \right) - 15 \left(-4556250 (-Y[6] + Y[7])^5 \left(-\frac{2}{5} + Y[7]^3 \right)^2 + 4556250 (-Y[7] + Y[8])^5 \left(-\frac{7}{15} + Y[8]^3 \right)^2 \right) == 0,$$

$$68343750Y[8]^2 (-Y[7] + Y[8])^6 \left(-\frac{7}{15} + Y[8]^3 \right) - 15 \left(-4556250 (-Y[7] + Y[8])^5 \left(-\frac{7}{15} + Y[8]^3 \right)^2 + 4556250 (-Y[8] + Y[9])^5 \left(-\frac{8}{15} + Y[9]^3 \right)^2 \right) == 0,$$

$$68343750Y[9]^2 (-Y[8] + Y[9])^6 \left(-\frac{8}{15} + Y[9]^3 \right) - 15 \left(-4556250 (-Y[8] + Y[9])^5 \left(-\frac{8}{15} + Y[9]^3 \right)^2 + 4556250 (-Y[9] + Y[10])^5 \left(-\frac{3}{5} + Y[10]^3 \right)^2 \right) == 0,$$

$$68343750Y[10]^2 (-Y[9] + Y[10])^6 \left(-\frac{3}{5} + Y[10]^3 \right) - 15 \left(-4556250 (-Y[9] + Y[10])^5 \left(-\frac{3}{5} + Y[10]^3 \right)^2 + 4556250 (-Y[10] + Y[11])^5 \left(-\frac{2}{3} + Y[11]^3 \right)^2 \right) == 0,$$

$$68343750Y[11]^2 (-Y[10] + Y[11])^6 \left(-\frac{2}{3} + Y[11]^3 \right) - 15 \left(-4556250 (-Y[10] + Y[11])^5 \left(-\frac{2}{3} + Y[11]^3 \right)^2 + 4556250 (-Y[11] + Y[12])^5 \left(-\frac{11}{15} + Y[12]^3 \right)^2 \right) == 0,$$

$$\begin{aligned}
& 68343750Y[12]^2(-Y[11] + Y[12])^6\left(-\frac{11}{15} + Y[12]^3\right) - 15\left(-4556250(-Y[11] + Y[12])^5\left(-\frac{11}{15} + Y[12]^3\right)^2\right. \\
& \quad \left.+ 4556250(-Y[12] + Y[13])^5\left(-\frac{4}{5} + Y[13]^3\right)^2\right) == 0, \\
& 68343750Y[13]^2(-Y[12] + Y[13])^6\left(-\frac{4}{5} + Y[13]^3\right) - 15\left(-4556250(-Y[12] + Y[13])^5\left(-\frac{4}{5} + Y[13]^3\right)^2\right. \\
& \quad \left.+ 4556250(-Y[13] + Y[14])^5\left(-\frac{13}{15} + Y[14]^3\right)^2\right) == 0, \\
& 68343750Y[14]^2(-Y[13] + Y[14])^6\left(-\frac{13}{15} + Y[14]^3\right) - 15\left(-4556250(-Y[13] + Y[14])^5\left(-\frac{13}{15} + Y[14]^3\right)^2\right. \\
& \quad \left.+ 4556250(-Y[14] + Y[15])^5\left(-\frac{14}{15} + Y[15]^3\right)^2\right) == 0, \\
& 68343750Y[15]^2(-Y[14] + Y[15])^6\left(-\frac{14}{15} + Y[15]^3\right) - 15\left(-4556250(-Y[14] + Y[15])^5\left(-\frac{14}{15} + Y[15]^3\right)^2\right. \\
& \quad \left.+ 4556250(-Y[15] + Y[16])^5(-1 + Y[16]^3)^2\right) == 0, Y[1] == 0, Y[16] == 1\}
\end{aligned}$$

Obj = Total[Table[(Sistema[[i, 1]] - Sistema[[i, 2]])^2, {i, 1, n + 1}]]

$$\begin{aligned}
& Y[1]^2 + \left(68343750Y[2]^2(-Y[1] + Y[2])^6\left(-\frac{1}{15} + Y[2]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[1] + Y[2])^5\left(-\frac{1}{15} + Y[2]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[2] + Y[3])^5\left(-\frac{2}{15} + Y[3]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[3]^2(-Y[2] + Y[3])^6\left(-\frac{2}{15} + Y[3]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[2] + Y[3])^5\left(-\frac{2}{15} + Y[3]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[3] + Y[4])^5\left(-\frac{1}{5} + Y[4]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[4]^2(-Y[3] + Y[4])^6\left(-\frac{1}{5} + Y[4]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[3] + Y[4])^5\left(-\frac{1}{5} + Y[4]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[4] + Y[5])^5\left(-\frac{4}{15} + Y[5]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[5]^2(-Y[4] + Y[5])^6\left(-\frac{4}{15} + Y[5]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[4] + Y[5])^5\left(-\frac{4}{15} + Y[5]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[5] + Y[6])^5\left(-\frac{1}{3} + Y[6]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[6]^2(-Y[5] + Y[6])^6\left(-\frac{1}{3} + Y[6]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[5] + Y[6])^5\left(-\frac{1}{3} + Y[6]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[6] + Y[7])^5\left(-\frac{2}{5} + Y[7]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[7]^2(-Y[6] + Y[7])^6\left(-\frac{2}{5} + Y[7]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[6] + Y[7])^5\left(-\frac{2}{5} + Y[7]^3\right)^2\right.\right. \\
& \quad \left.\left.+ 4556250(-Y[7] + Y[8])^5\left(-\frac{7}{15} + Y[8]^3\right)^2\right)\right)^2 \\
& + \left(68343750Y[8]^2(-Y[7] + Y[8])^6\left(-\frac{7}{15} + Y[8]^3\right)\right. \\
& \quad \left.- 15\left(-4556250(-Y[7] + Y[8])^5\left(-\frac{7}{15} + Y[8]^3\right)^2\right)\right)
\end{aligned}$$

$$\begin{aligned}
& +4556250(-Y[8] + Y[9])^5 \left(-\frac{8}{15} + Y[9]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[9]^2(-Y[8] + Y[9])^6 \left(-\frac{8}{15} + Y[9]^3\right) \right. \\
& - 15 \left(-4556250(-Y[8] + Y[9])^5 \left(-\frac{8}{15} + Y[9]^3\right)^2 \right. \\
& \left. +4556250(-Y[9] + Y[10])^5 \left(-\frac{3}{5} + Y[10]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[10]^2(-Y[9] + Y[10])^6 \left(-\frac{3}{5} + Y[10]^3\right) \right. \\
& - 15 \left(-4556250(-Y[9] + Y[10])^5 \left(-\frac{3}{5} + Y[10]^3\right)^2 \right. \\
& \left. +4556250(-Y[10] + Y[11])^5 \left(-\frac{2}{3} + Y[11]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[11]^2(-Y[10] + Y[11])^6 \left(-\frac{2}{3} + Y[11]^3\right) \right. \\
& - 15 \left(-4556250(-Y[10] + Y[11])^5 \left(-\frac{2}{3} + Y[11]^3\right)^2 \right. \\
& \left. +4556250(-Y[11] + Y[12])^5 \left(-\frac{11}{15} + Y[12]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[12]^2(-Y[11] + Y[12])^6 \left(-\frac{11}{15} + Y[12]^3\right) \right. \\
& - 15 \left(-4556250(-Y[11] + Y[12])^5 \left(-\frac{11}{15} + Y[12]^3\right)^2 \right. \\
& \left. +4556250(-Y[12] + Y[13])^5 \left(-\frac{4}{5} + Y[13]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[13]^2(-Y[12] + Y[13])^6 \left(-\frac{4}{5} + Y[13]^3\right) \right. \\
& - 15 \left(-4556250(-Y[12] + Y[13])^5 \left(-\frac{4}{5} + Y[13]^3\right)^2 \right. \\
& \left. +4556250(-Y[13] + Y[14])^5 \left(-\frac{13}{15} + Y[14]^3\right)^2 \Big)^2 \\
& + \left(68343750Y[14]^2(-Y[13] + Y[14])^6 \left(-\frac{13}{15} + Y[14]^3\right) \right. \\
& - 15 \left(-4556250(-Y[13] + Y[14])^5 \left(-\frac{13}{15} + Y[14]^3\right)^2 \right. \\
& \left. +4556250(-Y[14] + Y[15])^5 \left(-\frac{14}{15} + Y[15]^3\right)^2 \Big)^2 \\
& + (-1 + Y[16])^2 + \left(68343750Y[15]^2(-Y[14] + Y[15])^6 \left(-\frac{14}{15} + Y[15]^3\right) \right. \\
& - 15 \left(-4556250(-Y[14] + Y[15])^5 \left(-\frac{14}{15} + Y[15]^3\right)^2 \right. \\
& \left. +4556250(-Y[15] + Y[16])^5 (-1 + Y[16]^3)^2 \Big)^2
\end{aligned}$$

fObj = Function[Y, Obj];

ym = Max[Abs[ya], Abs[yb]];

vars = Table[Y[i], {i, 1, n + 1}];

{{Y[1] == ya, Y[n + 1] == yb}, Table[{0 ≤ Y[i] ≤ ym}, {i, 2, n}]};

cons = Flatten[%];

(*Método “Random Search”*)

sol1 = NMinimize[{fObj[vars], cons}, vars, Method → “RandomSearch”]// Timing

{13.907, {4.505772920546413×10⁻³³, {Y[1] → 0, Y[2] → 0.40548, Y[3] → 0.510873, Y[4] →

0.584804, Y[5] → 0.64366, Y[6] → 0.693387, Y[7] → 0.737589, Y[8] → 0.766182, Y[9] → 0.81118, Y[10] → 0.835427, Y[11] → 0.87358, Y[12] → 0.873547, Y[13] → 0.873524, Y[14] → 0.873507, Y[15] → 0.977265, Y[16] → 1.}}}

solnum1 = Table[Y[i], {i, 1, n + 1}]/.sol1[[2, 2]]

valorfunc1 = funcionald[abscissas, solnum1]

{0, 0.40548, 0.510873, 0.584804, 0.64366, 0.693387, 0.737589, 0.766182, 0.81118, 0.835427, 0.87358, 0.873547, 0.873524, 0.873507, 0.977265, 1.}
 $1.73015247761164 \times 10^{-7}$

(*Método “NelderMead”*)

sol2 = NMinimize[{fObj[vars], cons}, vars, Method → “NelderMead”]// Timing

{8.578, {2.4961950539161763 $\times 10^{-10}$, {Y[1] → 0, Y[2] → 0.40548, Y[3] → 0.396612, Y[4] → 0.39074, Y[5] → 0.393834, Y[6] → 0.3973, Y[7] → 0.400545, Y[8] → 0.403203, Y[9] → 0.400705, Y[10] → 0.398544, Y[11] → 0.401144, Y[12] → 0.403726, Y[13] → 0.406096, Y[14] → 0.408108, Y[15] → 0.406215, Y[16] → 1.}}}

solnum2 = Table[Y[i], {i, 1, n + 1}]/.sol2[[2, 2]]

valorfunc2 = funcionald[abscissas, solnum2]

{0, 0.40548, 0.396612, 0.39074, 0.393834, 0.3973, 0.400545, 0.403203, 0.400705, 0.398544, 0.401144, 0.403726, 0.406096, 0.408108, 0.406215, 1.}
 $3.118676622292057 \times 10^{-9}$

solnum1===solnum2

False

(*Método “Differential Evolution”*)

sol3 = NMinimize[{fObj[vars], cons}, vars, Method → “DifferentialEvolution”]// Timing

{8.172, {0., {Y[1] → 0, Y[2] → 0., Y[3] → 0., Y[4] → 0., Y[5] → 0., Y[6] → 0., Y[7] → 0., Y[8] → 0., Y[9] → 0., Y[10] → 0., Y[11] → 0., Y[12] → 0., Y[13] → 0., Y[14] → 0., Y[15] → 0., Y[16] → 1.}}}

solnum3 = Table[Y[i], {i, 1, n + 1}]/.sol3[[2, 2]]

valorfunc3 = funcionald[abscissas, solnum3]

{0, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.}
0.

solnum1===solnum3

solnum2===solnum3

False

False

(*Método “Simulated Annealing”*)

sol4 = NMinimize[{fObj[vars], cons}, vars, Method → “SimulatedAnnealing”]// Timing

{5.047, {1.6475888010043065 $\times 10^{-32}$, {Y[1] → 0, Y[2] → 0.0000229601, Y[3] → 0.510873, Y[4] → 0.584804, Y[5] →

solnum4 = Table[Y[i], {i, 1, n + 1}]/.sol4[[2, 2]]

valorfunc4 = funcionald[abcissas, solnum4]

{0, 0.0000229601, 0.510873, 0.584804, 0.64366, 0.693361, 0.736806, 0.775656, 0.810994, 0.84412, 0.866262,
0.90178, 0.928318, 0.95342, 0.977265, 1.}

2.6872916652232924 $\times 10^{-8}$

solnum1===solnum4

solnum2===solnum4

solnum3===solnum4

False

False

False

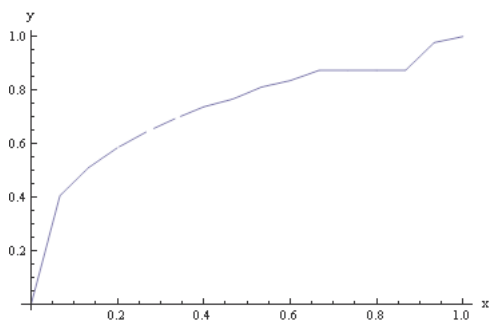
(*método random search*)

y1[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[

{X[i], solnum1[[i]]}, {X[i + 1], solnum1[[i + 1]]}], x], x], X[i] <= x < X[i + 1]], {i, 1, n}]]

$$\left\{ \begin{array}{ll} 0. + 6.0822x & 0 \leq x < \frac{1}{15} \\ 0.300087 + 1.58089x & \frac{1}{15} \leq x < \frac{2}{15} \\ 0.363012 + 1.10896x & \frac{2}{15} \leq x < \frac{1}{5} \\ 0.408235 + 0.882841x & \frac{1}{5} \leq x < \frac{4}{15} \\ 0.444748 + 0.745917x & \frac{4}{15} \leq x < \frac{1}{3} \\ 0.472379 + 0.663024x & \frac{1}{3} \leq x < \frac{2}{5} \\ 0.566034 + 0.428887x & \frac{2}{5} \leq x < \frac{7}{15} \\ 0.451194 + 0.674973x & \frac{7}{15} \leq x < \frac{8}{15} \\ 0.6172 + 0.363711x & \frac{8}{15} \leq x < \frac{3}{5} \\ 0.492047 + 0.5723x & \frac{3}{5} \leq x < \frac{2}{3} \\ 0.873912 - 0.00049785x & \frac{2}{3} \leq x < \frac{11}{15} \\ 0.873802 - 0.000347099x & \frac{11}{15} \leq x < \frac{4}{5} \\ 0.873729 - 0.000256604x & \frac{4}{5} \leq x < \frac{13}{15} \\ -0.475344 + 1.55637x & \frac{13}{15} \leq x < \frac{14}{15} \\ 0.658972 + 0.341028x & \frac{14}{15} \leq x < 1 \end{array} \right.$$

Plot[y1[x], {x, xa, xb}, AxesLabel → {"x", "y"}]



iy1 = Integrate[f[x, y1[x], D[y1[x], x]], {x, xa, xb}]

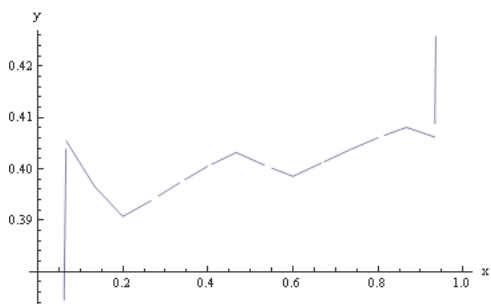
1.15649

(*método nelder mead*)

y2[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[
{X[i], solnum2[[i]]}, {X[i + 1], solnum2[[i + 1]]}], x], x], X[i] <= x < X[i + 1]}, {i, 1, n}]]

$$\left\{ \begin{array}{ll} 0. + 6.0822x & 0 \leq x < \frac{1}{15} \\ 0.414348 - 0.133019x & \frac{1}{15} \leq x < \frac{2}{15} \\ 0.408357 - 0.0880871x & \frac{2}{15} \leq x < \frac{1}{5} \\ 0.381456 + 0.0464174x & \frac{1}{5} \leq x < \frac{4}{15} \\ 0.379971 + 0.051989x & \frac{4}{15} \leq x < \frac{1}{3} \\ 0.381075 + 0.0486762x & \frac{1}{3} \leq x < \frac{2}{5} \\ 0.384597 + 0.0398699x & \frac{2}{5} \leq x < \frac{7}{15} \\ 0.42069 - 0.0374718x & \frac{7}{15} \leq x < \frac{8}{15} \\ 0.417993 - 0.0324141x & \frac{8}{15} \leq x < \frac{3}{5} \\ 0.375142 + 0.0390034x & \frac{3}{5} \leq x < \frac{2}{3} \\ 0.375324 + 0.0387306x & \frac{2}{3} \leq x < \frac{11}{15} \\ 0.37766 + 0.0355457x & \frac{11}{15} \leq x < \frac{4}{5} \\ 0.381957 + 0.0301741x & \frac{4}{5} \leq x < \frac{13}{15} \\ 0.432718 - 0.0283962x & \frac{13}{15} \leq x < \frac{14}{15} \\ -7.90678 + 8.90678x & \frac{14}{15} \leq x < 1 \end{array} \right.$$

Plot[y2[x], {x, xa, xb}, AxesLabel → {"x", "y"}]



iy2 = Integrate[f[x, y2[x], D[y2[x], x]], {x, xa, xb}]

12416.2

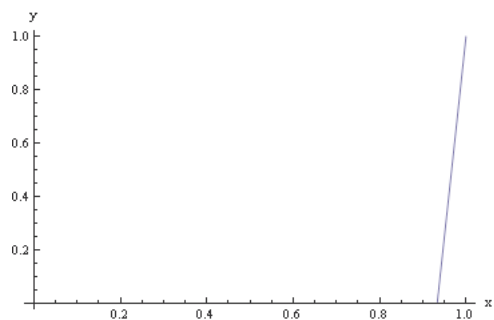
(*método differential evolution*)

y3[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[

{{X[i], solnum3[[i]]}, {X[i + 1], solnum3[[i + 1]]}}, x], x], X[i] <= x < X[i + 1]}, {i, 1, n}]]

$$\left\{ \begin{array}{ll} 0. + 0.x & 0 \leq x < \frac{1}{15} \parallel \frac{1}{15} \leq x < \frac{2}{15} \parallel \frac{2}{15} \leq x < \frac{1}{5} \parallel \frac{1}{5} \leq x < \frac{4}{15} \parallel \frac{4}{15} \leq x < \frac{1}{3} \parallel \\ & \frac{1}{3} \leq x < \frac{2}{5} \parallel \frac{2}{5} \leq x < \frac{7}{15} \parallel \frac{7}{15} \leq x < \frac{8}{15} \parallel \frac{8}{15} \leq x < \frac{3}{5} \parallel \frac{3}{5} \leq x < \frac{2}{3} \parallel \\ & \frac{2}{3} \leq x < \frac{11}{15} \parallel \frac{11}{15} \leq x < \frac{4}{5} \parallel \frac{4}{5} \leq x < \frac{13}{15} \parallel \frac{13}{15} \leq x < \frac{14}{15} \\ -14. + 15.x & \frac{14}{15} \leq x < 1 \end{array} \right.$$

Plot[y3[x], {x, xa, xb}, AxesLabel → {"x", "y"}]



iy3 = Integrate[f[x, y3[x], D[y3[x], x]], {x, xa, xb}]
 443732.

(*método simulated annealing*)

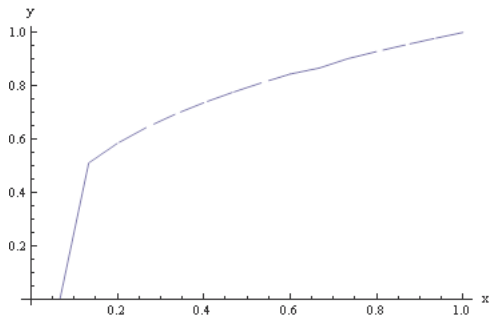
y4[x_] = Piecewise[Table[{Collect[InterpolatingPolynomial[
{X[i], solnum4[[i]]}, {X[i + 1], solnum4[[i + 1]]}, x], x], X[i] <= x < X[i + 1]}, {i, 1, n}]]

$$\left\{ \begin{array}{ll} 0. + 0.000344402x & 0 \leq x < \frac{1}{15} \\ -0.510827 + 7.66275x & \frac{1}{15} \leq x < \frac{2}{15} \\ 0.363012 + 1.10896x & \frac{2}{15} \leq x < \frac{1}{5} \\ 0.408235 + 0.882841x & \frac{1}{5} \leq x < \frac{4}{15} \\ 0.444853 + 0.745525x & \frac{4}{15} \leq x < \frac{1}{3} \\ 0.476136 + 0.651675x & \frac{1}{3} \leq x < \frac{2}{5} \\ 0.50371 + 0.58274x & \frac{2}{5} \leq x < \frac{7}{15} \\ 0.528286 + 0.530078x & \frac{7}{15} \leq x < \frac{8}{15} \\ 0.545985 + 0.496893x & \frac{8}{15} \leq x < \frac{3}{5} \\ 0.644846 + 0.332123x & \frac{3}{5} \leq x < \frac{2}{3} \\ 0.511084 + 0.532767x & \frac{2}{3} \leq x < \frac{11}{15} \\ 0.609861 + 0.39807x & \frac{11}{15} \leq x < \frac{4}{5} \\ 0.627097 + 0.376526x & \frac{4}{5} \leq x < \frac{13}{15} \\ 0.643431 + 0.35768x & \frac{13}{15} \leq x < \frac{14}{15} \\ 0.658972 + 0.341028x & \frac{14}{15} \leq x < 1 \end{array} \right.$$

Plot[y4[x], {x, xa, xb}, AxesLabel → {"x", "y"}]

iy4 = Integrate[f[x, y4[x], D[y4[x], x]], {x, xa, xb}]
 66.2651

qual o método que apresenta a melhor solução do sistema?



função objectivo do Problema de Programação Não Linear

```
ValorObjMin = Min[sol1[[2, 1]], sol2[[2, 1]], sol3[[2, 1]], sol4[[2, 1]]
MetfObjMin = Switch[ValorObjMin, sol1[[2, 1]], Print["Random Search"],
sol2[[2, 1]], Print["Nelder Mead"], sol3[[2, 1]], Print["Differential Evolution"], sol4[[2, 1]],
Print["Simulated Annealing"]]
SolfObjMin = Switch[ValorObjMin, sol1[[2, 1]], solnum1, sol2[[2, 1]],
solnum2, sol3[[2, 1]], solnum3, sol4[[2, 1]], solnum4];
0.
```

Differential Evolution

qual o método que apresenta a melhor solução do problema inicial?
valor do funcional discreto (somatório)

```
ValorFuncionalMin = Min[valorfunc1, valorfunc2, valorfunc3, valorfunc4]
SolFuncionalMin = Switch[ValorFuncionalMin, valorfunc1, Print["Random Search"],
valorfunc2, Print["Nelder Mead"], valorfunc3, Print["Differential Evolution"],
valorfunc4, Print["Simulated Annealing"]]
SolNumMin = Switch[ValorFuncionalMin, valorfunc1, solnum1,
valorfunc2, solnum2, valorfunc3, solnum3, valorfunc4, solnum4];
0.
```

Differential Evolution

qual o método que apresenta a melhor solução do problema inicial?
valor do funcional continuo

```
ValorIntegralMin = Min[iy1, iy2, iy3, iy4]
SolIntegralMin = Switch[ValorIntegralMin, iy1, Print["Random Search"], iy2,
Print["Nelder Mead"], iy3, Print["Differential Evolution"], iy4, Print["Simulated Annealing"]]
FuncIntMin[x_] = Switch[ValorIntegralMin, iy1, y1[x], iy2, y2[x], iy3, y3[x], iy4, y4[x]];
```

1.15649

Random Search

gráfico da aproximação de integral mínimo

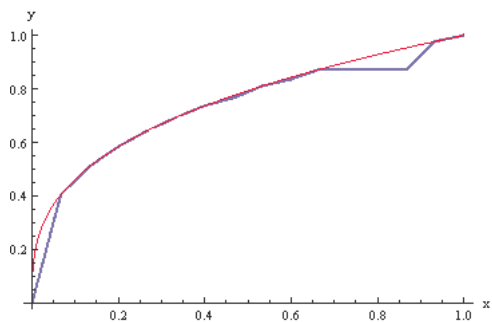
```
gmin = Plot[FuncIntMin[x], {x, xa, xb}, AxesLabel → {"x", "y"}, PlotStyle → Thick];
```

solução ótima para problema:

$$y = x^{1/3}$$

representação gráfica

```
gopt = Plot[x1/3, {x, 0, 1}, AxesLabel → {"x", "y"}, ColorFunction → Function[{x, y}, Red]];
Show[gmin, gopt]
```



Apêndice C

Código C++ usado na ferramenta OC

C.1 Problema da Braquistócrona

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//-----
char probName[] = "Braquistocrona";
int      ix = 2    ; // dimensao da variavel y(x) -> trajetoria
int      iu = 1    ; // dimensao da variavel u=y' -> controle
int      is = 0    ; // dimensao da variavel s
int      eTerm0 = 0 ; // n° restricoes '=' no inicio
int      lTerm0 = 0 ; // n° restricoes desigualdade no inicio
int      eTerm1 = 1 ; // n° restricoes '=' no fim
int      lTerm1 = 0 ; // n° restricoes desigualdade no fim
int      eTraj = 0  ; // n° restricoes '=' em cada ponto da particao
//de [xa,xb] excepto extremos
int      lTraj = 0  ; // n° restricoes desigualdade em cada ponto da
//particao de [xa,xb] excepto extremos
int      fixTime = 1 ; // 1 se tempo fixo e 0 se tempo não é fixo
double    t0 = 0    ; // xa
double    t1 = 10   ; // xb
int      nx = 3     ; // n° de partes em que o intervalo e dividido
int      uInt = 1    ; // multiplicador (multiplicidade dos u(t))
//-----
int      uType = 1    ;
```

```

int      scheme = 1    ;
int      pMult = 10.0;
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// sistema de controlo - y'(x) e u'(x)
ocp_f (double *x, double *u, double *s, double t, double *f)
{
f[1] = u[1];
f[2] = sqrt(1+pow(u[1],2))/((10-x[1]>1e-8)?sqrt(10-x[1]):0.0001);
}
//-----
extern "C" __declspec(dllexport) double __cdecl
//----->>

// expressao a minimizar - u(xb)
ocp_crit (double *xT, double *s)
{
return xT[2];
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// ponto inicial da trajetoria
ocp_x0 (double *x0, double *s)
{
x0[1] = 10; // y(xa)=ya
x0[2] = 0; // u(xa)=0
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no ponto inicial da trajetoria
ocp_EQ_t0 (double *x0, double *s, double *r)
{}
//-----

```

```

extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_LE_t0 (double *x0, double *s, double *r)
{
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no ponto final da trajetoria
ocp_EQ_T (double *xT, double *s, double *r)
{
r[1] = xT[1] - 0; // y(xb)-yb
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no intervalo ]xa,xb[
ocp_LE_T (double *xT, double *s, double *r)
{
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_EQ_tr (double *x, double *u, double t, double *r)
{
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_LE_tr (double *x, double *u, double t, double *r)
{
//-----
extern "C" __declspec(dllexport) int __cdecl
//----->>

ocp_u0 (double *u, int k)
{

```

```

int nu = nx / uInt + 1;
u[1] = 0;
return 1;
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_uMinMax (double *uMin, double *uMax)
{
uMin[1] = -20;
uMax[1] = 20;
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_s0 (double *s)
{}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_sMinMax (double *sMin, double *sMax)
{}

//=====
//      !!! Do not change the following lines !!!
//=====
typedef struct {
int _vers;
int _pFixTime;
int _ix;
int _iu;
int _is;
int _eTerm0;
int _lTerm0;
int _eTerm1;
int _lTerm1;

```



```

int _eTraj;
int _lTraj;
int _uType;
int _scheme;
int _nx;
int _uInt;
double _t0;
double _t1;
double _pMult;
char* _probName;
} ocp_Param;
//-----
ocp_Param PP;
//-----
//-----

extern "C" __declspec(dllexport)
void* __cdecl ocp_param(void)
{
PP._vers = 1;
PP._pFixTime = fixTime;
PP._ix = ix;
PP._iu = iu;
PP._is = is;
PP._eTerm0 = eTerm0;
PP._lTerm0 = lTerm0;
PP._eTerm1 = eTerm1;
PP._lTerm1 = lTerm1;
PP._eTraj = eTraj;
PP._lTraj = lTraj;
PP._uType = uType;
PP._scheme = scheme;
PP._nx = nx;
PP._uInt = uInt;
PP._t0 = t0;
PP._t1 = t1;
PP._pMult = pMult;
PP._probName = probName;
return ((void *) &PP);
}

```

```
//-----  
extern "C" __declspec(dllexport)  
void __cdecl ocp_nxuIntReset(int nxn, int uIntn)  
{  
    nx    = nxn;  
    uInt  = uIntn;  
}  
//=====
```

C.2 Problema com Fenómeno de Lavrentiev

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//-----
char probName[] = "Lavrentiev";
int      ix = 2    ; // dimensao da variavel y(x) -> trajetoria
int      iu = 1    ; // dimensao da variavel u=y' -> controlo
int      is = 0    ; // dimensao da variavel s
int      eTerm0 = 0 ; // nº restricoes '=' no inicio
int      lTerm0 = 0 ; // nº restricoes desigualdade no inicio
int      eTerm1 = 1 ; // nº restricoes '=' no fim
int      lTerm1 = 0 ; // nº restricoes desigualdade no fim
int      eTraj = 0  ; // nº restricoes '=' em cada ponto da particao
// de [xa,xb] excepto extremos
int      lTraj = 0  ; // nº restricoes desigualdade em cada ponto da
// particao de [xa,xb] excepto extremos
int      fixTime = 1 ; // 1 se tempo fixo e 0 se tempo não é fixo
double    t0 = 0    ; // xa
double    t1 = 1    ; // xb
int      nx = 3     ; // nº de partes em que o intervalo e dividido
int      uInt = 1    ; // multiplicador (multiplicidade dos u(t))
//-----
int      uType = 1   ;
int      scheme = 1   ;
int      pMult = 10.0;
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// sistema de controlo - y'(x) e u'(x)
ocp_f (double *x, double *u, double *s, double t, double *f)
{
f[1] = u[1];
f[2] = pow(pow(x[1],3)-t,2)*pow(u[1],6);
}
//-----
extern "C" __declspec(dllexport) double __cdecl
```

```

//----->>

// expressao a minimizar - u(1)
ocp_crit (double *xT, double *s)
{
return xT[2];
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// ponto inicial da trajetoria
ocp_x0 (double *x0, double *s)
{
x0[1] = 0; // y(xa)=ya
x0[2] = 1; // u(xa)=0
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no ponto inicial da trajetoria
ocp_EQ_t0 (double *x0, double *s, double *r)
{}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_LE_t0 (double *x0, double *s, double *r)
{}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no ponto final da trajetoria
ocp_EQ_T (double *xT, double *s, double *r)
{
r[1] = xT[1] - 1; // y(xb)-yb
}

```

```

//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

// restricoes no intervalo ]xa,xb[
ocp_LE_T (double *xT, double *s, double *r)
{}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_EQ_tr (double *x, double *u, double t, double *r)
{}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_LE_tr (double *x, double *u, double t, double *r)
{}
//-----
extern "C" __declspec(dllexport) int __cdecl
//----->>

ocp_u0 (double *u, int k)
{ int nu = nx / uInt + 1;

u[1] = 0;
return 1;
}
//-----
extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_uMinMax (double *uMin, double *uMax)
{
uMin[1] = -20;
uMax[1] = 20;
}
//-----

```

```

extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_s0 (double *s)
{}
//-----

extern "C" __declspec(dllexport) void __cdecl
//----->>

ocp_sMinMax (double *sMin, double *sMax)
{}

//=====
//      !!! Do not change the following lines !!!
//=====
typedef struct {
int _vers;
int _pFixTime;
int _ix;
int _iu;
int _is;
int _eTerm0;
int _lTerm0;
int _eTerm1;
int _lTerm1;
int _eTraj;
int _lTraj;
int _uType;
int _scheme;
int _nx;
int _uInt;
double _t0;
double _t1;
double _pMult;
char* _probName;
} ocp_Param;
//-----
ocp_Param PP;
//-----

```

```

//-----
extern "C" __declspec(dllexport)
void* __cdecl ocp_param(void) {
PP._vers = 1;
PP._pFixTime = fixTime;
PP._ix = ix;
PP._iu = iu;
PP._is = is;
PP._eTerm0 = eTerm0;
PP._lTerm0 = lTerm0;
PP._eTerm1 = eTerm1;
PP._lTerm1 = lTerm1;
PP._eTraj = eTraj;
PP._lTraj = lTraj;
PP._uType = uType;
PP._scheme = scheme;
PP._nx = nx;
PP._uInt = uInt;
PP._t0 = t0;
PP._t1 = t1;
PP._pMult = pMult;
PP._probName = probName;
return ((void *) &PP);
}
//-----
extern "C" __declspec(dllexport)
void __cdecl ocp_nxuIntReset(int nxn, int uIntn) {
nx = nxn;
uInt = uIntn;
}
//=====

```


Apêndice D

Soluções encontradas pela ferramenta OC

D.1 Problema da Braquistócrona

Tabela D.1: Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).

MPM	MED		Solução
US	Euler	PL	{10.000000, 9.988336, 9.703641, 8.840579, 5.909270, 2.771670, 1.669849, 0.932133, 0.435708, 0.065031, -0.208611, -0.405196, -0.537323, -0.605404, -0.609577, -0.556104, -0.458293, -0.345295, -0.231292, -0.117364, -0.003283}
		PC	{10.000000, 9.988454, 9.731829, 9.139977, 8.113098, 5.159688, 3.330584, 2.299567, 1.591214, 1.072535, 0.670855, 0.344045, 0.090691, -0.109346, -0.250889, -0.334598, -0.366738, -0.345481, -0.267891, -0.147836, -0.004324}
	R-K 2	PL	{10.000000, 5.755559, 5.363224, 4.613933, 4.242782, 3.700543, 3.366575, 2.931252, 2.633842, 2.272040, 2.008192, 1.703410, 1.471158, 1.212526, 1.009336, 0.790650, 0.614271, 0.430495, 0.280846, 0.127761, 0.003822}
		PC	{10.000000, 4.635664, 4.169098, 3.741865, 3.349466, 2.988389, 2.655241, 2.348057, 2.063973, 1.800860, 1.557510, 1.332422, 1.124630, 0.933444, 0.758352, 0.598834, 0.454191, 0.323420, 0.205140, 0.098511, 0.002760}

Tabela D.2: Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).

MPM	MED		Solução
DS 1	Euler	PL	{10.000000, 9.988460, 9.732290, 9.148212, 8.405778, 7.621869, 6.856510, 6.134017, 5.459573, 4.831820, 4.246735, 3.700017, 3.186665, 2.703944, 2.250091, 1.822140, 1.418170, 1.036495, 0.675065, 0.333033, 0.008744}
		PC	{10.000000, 9.988460, 9.732290, 9.148212, 8.405778, 7.621869, 6.856510, 6.134017, 5.459573, 4.831820, 4.246735, 3.700017, 3.186665, 2.703944, 2.250091, 1.822140, 1.418170, 1.036495, 0.675065, 0.333033, 0.008744}
	R-K 2	PL	{10.000000, 5.792461, 5.388141, 4.657986, 4.271368, 3.742405, 3.395137, 2.968130, 2.660805, 2.302945, 2.032663, 1.727543, 1.491056, 1.231073, 1.024821, 0.802318, 0.625316, 0.437443, 0.287180, 0.129394, 0.003897}
		PC	{10.000000, 4.705851, 4.240514, 3.814312, 3.422022, 3.059291, 2.723380, 2.411451, 2.122282, 1.854575, 1.606288, 1.375605, 1.162476, 0.965872, 0.785034, 0.619032, 0.467507, 0.331014, 0.208084, 0.098801, 0.002795}
DS 2	Euler	PL	{10.000000, 9.988459, 9.732368, 9.148734, 8.407268, 7.624793, 6.860348, 6.135869, 5.459611, 4.829598, 4.242766, 3.693876, 3.179474, 2.696586, 2.242520, 1.815665, 1.413037, 1.032413, 0.672565, 0.331738, 0.008740}
		PC	{10.000000, 9.988459, 9.732368, 9.148734, 8.407268, 7.624793, 6.860348, 6.135869, 5.459611, 4.829598, 4.242766, 3.693876, 3.179474, 2.696586, 2.242520, 1.815665, 1.413037, 1.032413, 0.672565, 0.331738, 0.008740}
	R-K 2	PL	{10.000000, 5.792382, 5.386942, 4.657395, 4.270173, 3.740644, 3.394516, 2.969580, 2.662323, 2.304436, 2.033899, 1.728206, 1.490978, 1.230327, 2.242520, 1.815665, 1.413037, 1.032413, 0.672565, 0.331738, 0.008740}
		PC	{10.000000, 4.715657, 4.249045, 3.821651, 3.428284, 3.065060, 2.728867, 2.417498, 2.128786, 1.860880, 1.612339, 1.382254, 1.168855, 0.971559, 0.789347, 0.622349, 0.469965, 0.332514, 0.209220, 0.099463, 0.002820}
RS	Euler	PL	{10.000000, 9.988458, 9.732377, 9.147091, 8.403722, 7.623440, 6.845750, 6.114098, 5.457095, 4.818923, 4.234377, 3.680296, 3.170746, 2.690312, 2.244798, 1.822899, 1.419535, 1.038041, 0.672318, 0.332416, 0.008702}
		PC	{10.000000, 9.988408, 9.727433, 9.161244, 8.422860, 7.607729, 6.733391, 6.111990, 5.535410, 4.611601, 4.088468, 3.582729, 3.076121, 2.597590, 2.186961, 1.754145, 1.372286, 1.019356, 0.650057, 0.312927, 0.009560}
	R-K 2	PL	{10.000000, 5.747191, 5.351358, 4.600708, 4.227917, 3.681968, 3.351843, 2.908511, 2.625907, 2.255658, 2.005482, 1.690510, 1.469623, 1.199216, 1.007391, 0.776618, 0.608488, 0.420853, 0.275461, 0.119485, 0.003868}
		PC	{10.000000, 4.708260, 4.242005, 3.815088, 3.422137, 3.059208, 2.723180, 2.411605, 2.122459, 1.854153, 1.605318, 1.374856, 1.161750, 0.965169, 0.784418, 0.618865, 0.467968, 0.331295, 0.208452, 0.099042, 0.002793}

Tabela D.3: Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).

MPM	MED		Solução
CG	Euler	PL	{10.000000, 9.988458, 9.732341, 9.148425, 8.406231, 7.622636, 6.856965, 6.133300, 5.457667, 4.828665, 4.242514, 3.695029, 3.182310, 2.700928, 2.247942, 1.820842, 1.417494, 1.036073, 0.675013, 0.332962, 0.008747}
		PC	{10.000000, 9.988458, 9.732341, 9.148425, 8.406231, 7.622636, 6.856965, 6.133300, 5.457667, 4.828665, 4.242514, 3.695029, 3.182310, 2.700928, 2.247942, 1.820842, 1.417494, 1.036073, 0.675013, 0.332962, 0.008747}
	R-K 2	PL	{10.000000, 5.796500, 5.390215, 4.664381, 4.274809, 3.747934, 3.397757, 2.972591, 2.662802, 2.306164, 2.033630, 1.730388, 1.491552, 1.232240, 1.024699, 0.803171, 0.624707, 0.436719, 0.285643, 0.128537, 0.003883}
		PC	{10.000000, 4.708897, 4.242715, 3.815740, 3.422861, 3.059749, 2.723550, 2.411879, 2.122685, 1.854308, 1.605419, 1.374903, 1.161795, 0.965247, 0.784516, 0.618963, 0.468051, 0.331331, 0.208427, 0.099013, 0.002794}
NM	Euler	PL	{10.000000, 9.988477, 9.733502, 9.163995, 8.477871, 7.783251, 7.278255, 6.508288, 5.597911, 4.398964, 4.233971, 3.253427, 2.096437, 1.798959, 1.262582, 1.282262, 1.659590, 1.329463, 0.795952, 0.284717, 0.004483}
		PC	{10.000000, 9.988458, 9.732334, 9.148428, 8.406222, 7.622601, 6.857332, 6.133623, 5.457656, 4.828553, 4.242855, 3.694969, 3.182305, 2.700963, 2.247931, 1.820879, 1.417533, 1.035922, 0.675029, 0.332890, 0.008751}
	R-K 2	PL	{10.000000, 5.138522, -2.711172, -4.029433, -3.698507, -8.225042, -9.483711, -7.169755, -7.891304, -10.729986, -13.146559, -11.965729, -12.389639, -15.171217, -12.395373, -7.475894, -4.362363, -5.987098, -6.740944, -1.864680, -0.000018}
		PC	{10.000000, 6.123066, 1.775744, 0.837276, 0.259239, 3.999358, -1.948377, -11.942703, -6.597754, 2.504360, -5.198674, -13.915299, -4.163213, 5.798730, -4.080023, -7.017857, -12.454142, -2.764647, -12.615000, -5.713733, 0.004480}

D.2 Problema que exhibe o Fenómeno de Lavrentiev

Tabela D.4: Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).

MPM	MED		Solução
US	Euler	PL	{0.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000}
		PC	{0.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000}
	R-K 2	PL	{0.000000, 0.357064, 0.460808, 0.529498, 0.583288, 0.633518, 0.671994, 0.710141, 0.741861, 0.771412, 0.794548, 0.812560, 0.833406, 0.850526, 0.872025, 0.893294, 0.920095, 0.948545, 0.973182, 0.992095, 0.999999}
		PC	{0.000000, 0.125512, 0.717919, 0.706607, 0.693912, 0.680415, 0.683448, 0.703970, 0.737835, 0.770825, 0.799821, 0.824141, 0.843880, 0.861870, 0.879864, 0.898999, 0.921393, 0.946035, 0.968147, 0.988943, 1.000000 }
DS 1	Euler	PL	{0.000000, 0.453216, 0.464161, 0.568488, 0.584787, 0.643353, 0.667934, 0.702528, 0.735367, 0.771318, 0.795769, 0.823856, 0.847966, 0.870503, 0.888032, 0.905300, 0.924778, 0.945423, 0.971910, 0.991255, 0.999998}
		PC	{0.000000, 0.453216, 0.464161, 0.568488, 0.584787, 0.643353, 0.667934, 0.702528, 0.735367, 0.771318, 0.795769, 0.823856, 0.847966, 0.870503, 0.888032, 0.905300, 0.924778, 0.945423, 0.971910, 0.991255, 0.999998}
	R-K 2	PL	{0.000000, 0.040180, 0.460860, 0.621167, 0.636158, 0.654713, 0.671404, 0.688286, 0.705646, 0.721053, 0.739232, 0.753858, 0.773329, 0.787593, 0.809436, 0.823840, 0.848496, 0.863322, 0.895267, 0.910919, 0.999998}
		PC	{0.000000, 0.116402, 0.192378, 0.255069, 0.310536, 0.361332, 0.408821, 0.453851, 0.497001, 0.538689, 0.579245, 0.618943, 0.658021, 0.696711, 0.735244, 0.773887, 0.812980, 0.853008, 0.894794, 0.940220, 0.998478}
DS 2	Euler	PL	{0.000000, 0.368407, 0.573554, 0.572543, 0.584917, 0.649997, 0.668850, 0.704187, 0.745135, 0.763850, 0.791595, 0.821831, 0.845506, 0.871525, 0.888385, 0.918347, 0.928999, 0.945733, 0.965430, 1.009209, 1.000002}
		PC	{0.000000, 0.368407, 0.573554, 0.572543, 0.584917, 0.649997, 0.668850, 0.704187, 0.745135, 0.763850, 0.791595, 0.821831, 0.845506, 0.871525, 0.888385, 0.918347, 0.928999, 0.945733, 0.965430, 1.009209, 1.000002}

Tabela D.5: Soluções encontradas pela ferramenta OC para $n = 20$ usando diferentes métodos para resolver os problemas de Programação Matemática (MPM) e as Equações Diferenciais (MED).

MPM	MED		Solução
DS 2	R-K 2	PL	{0.000000, 0.038343, 0.460863, 0.629922, 0.644115, 0.658682, 0.672060, 0.687089, 0.703003, 0.717876, 0.735970, 0.748439, 0.765892, 0.779094, 0.801787, 0.815707, 0.842511, 0.856761, 0.952141, 0.969156, 0.999999}
		PC	{0.000000, 0.116402, 0.192381, 0.255071, 0.310540, 0.361333, 0.408820, 0.453852, 0.497002, 0.538693, 0.579251, 0.618947, 0.658022, 0.696711, 0.735244, 0.773889, 0.812979, 0.853003, 0.894793, 0.940219, 0.998477}
RS	Euler	PL	{0.000000, 0.368403, 0.464180, 0.531339, 0.610498, 0.629418, 0.669268, 0.705046, 0.736651, 0.776419, 0.794886, 0.817816, 0.843555, 0.866262, 0.903812, 0.908584, 0.952901, 0.947264, 1.014934, 1.008254, 1.000001}
		PC	{0.000000, 0.368402, 0.464161, 0.531293, 0.584853, 0.629923, 0.669482, 0.704743, 0.754933, 0.766297, 0.810071, 0.819314, 0.864370, 0.866190, 0.895593, 0.909209, 0.928304, 0.969065, 0.965484, 0.988580, 1.000000}
	R-K 2	PL	{0.000000, 0.116352, 0.193248, 0.255068, 0.311150, 0.361186, 0.409425, 0.453758, 0.497505, 0.538210, 0.579331, 0.618445, 0.657876, 0.695686, 0.734686, 0.772478, 0.811745, 0.850683, 0.893362, 0.936031, 0.998283}
		PC	{0.000000, 0.116403, 0.192381, 0.255073, 0.310541, 0.361334, 0.408822, 0.453853, 0.497002, 0.538692, 0.579249, 0.618947, 0.658026, 0.696714, 0.735246, 0.773889, 0.812981, 0.853005, 0.894794, 0.940220, 0.998479}
CG	Euler	PL	{0.000000, 0.620000, 0.623565, 0.603986, 0.595677, 0.639930, 0.678537, 0.710531, 0.745184, 0.778312, 0.806320, 0.829805, 0.849962, 0.869119, 0.890147, 0.914184, 0.940351, 0.965341, 0.983359, 0.989112, 0.999997}
		PC	{0.000000, 0.620000, 0.623565, 0.603986, 0.595677, 0.639930, 0.678537, 0.710531, 0.745184, 0.778312, 0.806320, 0.829805, 0.849962, 0.869119, 0.890147, 0.914184, 0.940351, 0.965341, 0.983359, 0.989112, 0.999997}
	R-K 2	PL	{0.000000, 0.023812, 0.461218, 0.558307, 0.595090, 0.633547, 0.659368, 0.690734, 0.712059, 0.735160, 0.756861, 0.777438, 0.792851, 0.812325, 0.834371, 0.852269, 0.874832, 0.892805, 0.922628, 0.941349, 1.000035}
		PC	{0.000000, 0.116385, 0.192334, 0.255019, 0.310494, 0.361291, 0.408776, 0.453802, 0.496950, 0.538641, 0.579204, 0.618909, 0.657994, 0.696686, 0.735220, 0.773866, 0.812964, 0.852992, 0.894784, 0.940222, 0.998461}
NM	Euler	PL	*
		PC	*
	R-K 2	PL	*
		PC	*

‘*’: Invalid floating point operation.

Bibliografia

- [1] Y. Bai and Z.-P. Li, *A Truncation Method for Detecting Singular Minimizers Involving the Lavrentiev Phenomenon*, Mathematical Models and Methods in Applied Sciences, Vol. 16 (2006), No.6, pp. 847–867.
- [2] Raktim Bhattacharya, *OPTRAGEN: A MATLAB Toolbox for Optimal Trajectory Generation*, 45th IEEE Conference on Decision and Control, San Diego, CA, 2006, pp. 6832–6836.
- [3] Pedro A. F. Cruz and Delfim F. M. Torres, *Evolution Strategies on Optimal Control*, Proc. Estonian Acad. Sci. Phys. Math., 2007, 56(4), pp. 299–309.
- [4] Gusein Sh. Guseinov, *Discrete Calculus of Variations*, Global Analysis and Applied Mathematics, American Institute of Physics Conference Proceedings, Melville, New York, 2004, pp. 170–176.
- [5] Claudio Padra, *The Beginnings of Variational Calculus and it's Early Relation With Numerical Methods*, Variational Formulations in Mechanics: Theory and Applications, E. Taroco, E.A. de Souza e A.A. Novotny (Eds.), CIMNE, Barcelona, Espanha, 2006.
- [6] Walter G. Kelley, Allan C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press, Inc., Boston, MA, 1991.
- [7] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, John Wiley & Sons, Inc., 1993.
- [8] Herman H. Goldstine, *A History of the Calculus of Variations from the 17th through the 19th Century*, Springer-Verlag, New York, 1980.
- [9] Elon Lages Lima, *Curso de Análise, Vol. 2*, Projecto Euclides, IMPA, Rio de Janeiro, 1995.
- [10] G. Smirnov, V. Bushenkov, *Curso de Optimização - Programação Matemática, Cálculo das Variações, Controlo Óptimo*, Escolar Editora, Lisboa, 2005.

- [11] Gilbert Strang, *Linear Algebra And Its Applications*, Harcourt Brace Jovanovich, Inc., 1988.
- [12] Delfim Torres, *Introdução à Optimização Dinâmica*, Universidade de Aveiro, Departamento de Matemática, 2005.
- [13] Robert Weinstock, *Calculus of Variations*, McGraw-Hill, USA, 1952.
- [14] Alessandro Ferriero, *The Lavrentiev phenomenon in the Calculus of Variations*, Dipartimento di Matematica ed Applicazioni, Università degli Studi di Milano-Bicocca, 2004.
- [15] Andreia M. F. Louro, *Computação Simbólica em Maple no Cálculo das Variações*, Dissertação de Mestrado, Departamento de Matemática, Universidade de Aveiro, 2006.
- [16] Cristiana J. S. da Silva, *Abordagens do Cálculo das Variações e Controlo Óptimo ao Problema de Newton de Resistência Mínima*, Departamento de Matemática, Universidade de Aveiro, 2005.
- [17] Delfim F. M. Torres, *Regularidade Lipschitziana dos minimizante no Cálculo das Variações e Controlo Óptimo*, Departamento de Matemática, Universidade de Aveiro, 1997.

Índice

- Cálculo das Variações, 75
 - forma geral, 2, 26
 - formulação de Lagrange, 27
 - método de Euler, 2, 14
 - método de Lagrange, 3
- Controlo Ótimo, 26, 76
- Desigualdade de Hölder, 59
- Diferenças Finitas de Euler, 6, 8, 14, 15
- Differential Evolution, 41, 48, 50, 62, 63, 76
- discretização
 - Euler, 2
 - Guseinov, 7, 12, 16, 43, 48, 50–53, 60, 72, 76
 - Smirnov-Bushenkov, 6
 - usual, 6, 8, 14, 19, 43, 48, 50–53, 60–62, 72, 73, 76
- Energia Cinética, 45
- Energia Mecânica, 45
- Energia Potencial, 45
- equação de Euler-Lagrange, 5
- equações de Euler, 3
- equações de Euler-Lagrange, 3, 6, 8, 13, 15, 16, 19, 21
- Exemplo de Manià, 57, 66, 72, 77
- extremal, 5, 11, 21
- extremal de Potryagin, 28
- Fenómeno de Lavrentiev, 57, 62, 66, 77
- forma de Bolza, 28
- forma de Mayer, 28, 54, 77
- formulação de Bolza, 31, 48, 60
- formulação de Mayer, 32, 48, 61
- função absolutamente contínua, 57, 58
- função Lipschitziana, 57, 59, 65
- funcional, 2, 26
- funcional discreto, 6, 7
- Hamiltoniano, 27
- Lagrangiano Contínuo, 2, 40
- Lagrangiano Discreto, 6, 38, 40
- Lei de Conservação de Energia, 45
- Mathematica, 37, 72, 75
- Nelder Mead, 40, 48, 50–52, 61, 62, 76
- OC, 32, 48, 54, 61, 66, 72, 73
- OPTRAGEN, 31, 48, 60
- Princípio do Máximo de Pontryagin, 27
- Problema da Braquistócrona, 1, 3, 42, 48, 54, 72, 77
- Problema de Resistência Mínima de Newton, 76
- Programação Não Linear, 40, 41, 48, 51, 61
- Random Search, 40, 48, 50–53, 62, 63, 72, 73, 76
- Simulated Annealing, 41, 48, 50, 61–63, 73, 76
- Sistema Adjunto, 28
- sistema de equações não lineares, 40, 41, 48
- Teorema do Valor Médio, 58